

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-315162

(43)Date of publication of application : 14.11.2000

(51)Int.Cl.

G06F 9/46

(21)Application number : 11-123675

(71)Applicant : MATSUSHITA ELECTRIC IND CO
LTD

(22)Date of filing : 30.04.1999

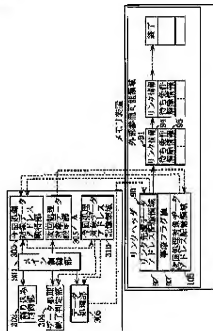
(72)Inventor : KAREKI MASAYOSHI

(54) CONTROL BLOCK PROCESSOR AND COMPUTER READABLE RECORDING MEDIUM
RECORDING CONTROL BLOCK PROCESSING PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To shorten interruption inhibition time in the processing of data having exclusive link structure and to shorten the extension of processing time to be required for system call when the interruption inhibition time is shortened.

SOLUTION: A text processing target data address storage area 105 for finding out and storing the address of data to be processed next is formed in a memory area allowed to be referred to from the external. When the link structure of link structure data being executed in system call processing interrupted by an interruption generated when interruption is temporarily permitted during the period of system call processing is changed by another system call processing, the system call processing part changing the link structure sets up the optimum data address of the changed link structure data in the storage area 105.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The link structure data which consist of two or more control block combined using the link information, The address storage region where the address of which control block is written in among two or more control block, The actuation which determines control block by which the address is written in the address storage region as a processing object this time, Control block of the link place of the control block is determined as a processing object next time. The processing-object decision means which repeats the actuation which writes the address of a processing object in an address storage region the next time concerned about all control block in link structure data, A processing means to perform processing about a processing object this time [this] whenever a processing object is determined this time, It has an authorization means to permit generating of interruption processing in a period after processing about a processing object is performed this time until a processing object is determined as a degree this time. Said processing-object decision means While it is started when interruption processing occurs in the period when interruption processing was permitted, and determining which control block in link structure data as a processing object this time in interruption processing It determines as a processing object the next time which should be processed in case processing of the side interrupted in another control block is resumed. [when resuming processing of the side which it has the reentrant routine which writes the address of a processing object in an address storage region the next time concerned, and was interrupted after interruption processing generating, and when interruption processing does not occur] The control-block processor characterized by determining control block by which the address is written in the address storage region as a processing object this time.

[Claim 2] It is the control-block processor according to claim 1 characterized by canceling which thing of link structure data among said two or more control block at the time of interruption processing activation, and for said reentrant routine determining control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region.

[Claim 3] When interruption processing is materialized [in / in any they are / a control-block processor] among two or more events, it is generated. To each of control block The task corresponds and it has the event information which shows the event to which the task corresponding to this is waiting for formation to control block. What was in agreement with the event it was notified in interruption processing among said two or more control block that the event information was After performing at the time of interruption processing activation, it is canceled of link structure data. Said reentrant routine The control-block processor according to claim 2 characterized by determining control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region.

[Claim 4] Said reentrant routine is a control-block processor according to claim 3 characterized by determining control block used as the link place of control block canceled in link structure data as a processing object next time, and writing the address in an address storage region.

[Claim 5] In the time of interruption processing activation, control block corresponding to which task is inserted in link structure data. Said reentrant routine It has the decision section which determines any of the task corresponding to control block and inserted control block for [which should be performed at the time of the restart to the interrupted side] a restart it should perform

first they are. The control-block processor according to claim 4 characterized by determining control block determined by the decision section as a processing object next time, and writing the address in an address storage region.

[Claim 6] It is a control-block processor given in any of claims 1-5 characterized by giving the priority based on the superiority or inferiority of the task corresponding to it to control block, and for said decision section measuring a priority by control block for [which should be performed at the time of the restart to the interrupted side] a restart, and control block for [which was inserted] insertion, and determining any of the task corresponding to any should be performed first they are.

[Claim 7] It is the control-block processor according to claim 6 characterized by for link structure data having the link header which shows control block corresponding to the task which should be performed to the beginning among two or more tasks, and for a reentrant routine determining control block shown in a link header among control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region.

[Claim 8] It is the control-block processor according to claim 1 which a control-block processor has two or more tasks and free memory block which is empty memory block, and each control block corresponds with each of two or more free memory block, and is characterized by equipping said processing means with the allocation section which assigns free memory block corresponding to control block which serves as a processing object this time to which task.

[Claim 9] It is the control-block processor according to claim 8 characterized by canceling which thing of link structure data among said two or more control block at the time of interruption processing activation, and for said reentrant routine determining control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region.

[Claim 10] Said reentrant routine is a control-block processor according to claim 9 characterized by determining control block used as the link place of control block canceled in link structure data as a processing object next time, and writing the address in an address storage region.

[Claim 11] In the time of said interruption processing activation, control block corresponding to which free memory block is inserted in link structure data. Said reentrant routine It has the decision section which determines any of free memory block corresponding to control block and inserted control block for [which should be assigned at the time of the restart of processing of the interrupted side] a restart should be first assigned to a task. A control-block processor given in any of claims 8-10 characterized by determining control block determined by the decision section as a processing object next time, and writing the address in an address storage region they are.

[Claim 12] It is the control-block processor according to claim 11 characterized by giving the priority based on the memory size of free memory block corresponding to it to control block, and for said decision section measuring a priority by control block for [which should be assigned to a task at the time of the restart to the interrupted side] a restart, and control block for [which was inserted] insertion, and determining any of free memory block corresponding to any should be first assigned to a task.

[Claim 13] It is the control-block processor according to claim 12 which link structure data have the link header which shows the address of control block corresponding to free memory block which should be assigned to the beginning among two or more free memory block at a task, and is characterized by the 1st allocation section writing the address of control block shown in a link header among control block which remained in link structure data, without being canceled in a global field.

[Claim 14] It is the record medium which recorded the control-block processing program and in which computer reading is possible. Said computer The link structure data which consist of two or more control block combined using the link information, It has the address storage region where the address of which control block is written in among two or more control block. Said control-block processing program The actuation which determines control block by which the address is written in the address storage region as a processing object this time, Control block of the link place of the control block concerned is determined as a processing object next time. The processing-object decision step which repeats the actuation which writes the address of a processing object in an address storage region the next time concerned about all control block in link structure data, The

processing step which performs processing about a processing object this time [this] whenever a processing object is determined this time. It has the authorization step which permits generating of interruption processing in a period after processing about a processing object is performed this time until a processing object is determined as a degree this time. While said processing-object decision step is started when interruption processing occurs in the period when interruption processing was permitted, and determining which control block in link structure data as a processing object this time in interruption processing It determines as a processing object the next time which should be processed in case processing of the side interrupted in another control block is resumed. [when resuming processing of the side which it has the reentrant routine which writes the address of a processing object in an address storage region the next time concerned, and was interrupted after interruption processing generating, and when interruption processing does not occur] The record medium which is characterized by determining control block by which the address is written in the address storage region as a processing object this time and in which computer reading is possible.

[Claim 15] It is the record medium which is characterized by canceling which thing of link structure data among said two or more control block at the time of interruption processing activation, and for said reentrant routine determining control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 14 is possible.

[Claim 16] When interruption processing is materialized [in / in any they are / the program of a control-block processing record medium] among two or more events, it is generated. To each of control block The task corresponds and it has the event information which shows the event to which the task corresponding to this is waiting for formation to control block. What was in agreement with the event it was notified in interruption processing among said two or more control block that the event information was After performing at the time of interruption processing activation, it is canceled of link structure data. Said reentrant routine The record medium which is characterized by determining control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 15 is possible.

[Claim 17] Said reentrant routine is a record medium which is characterized by determining control block used as the link place of control block canceled in link structure data as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 16 is possible.

[Claim 18] In the time of interruption processing activation, control block corresponding to which task is inserted in link structure data. Said reentrant routine It has the decision substep which determines any of the task corresponding to control block and inserted control block for [which should be performed at the time of the restart to the interrupted side] a restart it should perform first they are. The record medium which is characterized by determining control block determined by the decision substep as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 17 is possible.

[Claim 19] It is the record medium which can computer read the publication for any of claims 14-18 characterized by to give the priority based on the superiority or inferiority of the task corresponding to it to control block, and for said decision substep to measure a priority by control block for [which should be performed at the time of the restart to the interrupted side] a restart, and control block for [which was inserted] insertion, and to determine any of the task corresponding to any should be performed first their being.

[Claim 20] It is the record medium which is characterized by for link structure data having the link header which shows control block corresponding to the task which should be performed to the beginning among two or more tasks, and for a reentrant routine determining control block shown in a link header among control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 19 is possible.

[Claim 21] It is the record medium which the record medium in which computer reading is possible has two or more tasks and free memory block which is empty memory block, and each control block corresponds with each of two or more free memory block, and is characterized by said processing step consisting of an allocation substep which assigns free memory block corresponding

to control block which serves as a processing object this time to which task and in which computer reading according to claim 14 is possible.

[Claim 22] It is the record medium which is characterized by canceling which thing of link structure data among said two or more control block at the time of interruption processing activation, and for said reentrant routine determining control block which remained in link structure data, without being canceled as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 21 is possible.

[Claim 23] Said reentrant routine is a record medium which is characterized by determining control block used as the link place of control block canceled in link structure data as a processing object next time, and writing the address in an address storage region and in which computer reading according to claim 22 is possible.

[Claim 24] In the time of said interruption processing activation, control block corresponding to which free memory block is inserted in link structure data. Said reentrant routine It has the decision substep which determines any of free memory block corresponding to control block and inserted control block for [which should be assigned at the time of the restart of processing of the interrupted side] a restart should be first assigned to a task. The record medium which can computer read the publication for any of claims 21-23 characterized by determining control block determined by the decision substep as a processing object next time, and writing the address in an address storage region their being.

[Claim 25] It is the record medium which is characterized by giving the priority based on the memory size of free memory block corresponding to it to control block, and for said decision substep measuring a priority by control block for [which should be assigned to a task at the time of the restart to the interrupted side] a restart, and control block for [which was inserted] insertion, and determining any of free memory block corresponding to any should be first assigned to a task and in which computer reading according to claim 24 is possible.

[Claim 26] It is the record medium which link structure data have the link header which shows the address of control block corresponding to free memory block which should be assigned to the beginning among two or more free memory block at a task, and is characterized by a 1st allocation substep writing the address of control block shown in a link header among control block which remained in link structure data, without being canceled in a global field and in which computer reading according to claim 25 is possible.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the record medium which recorded the control-block processor which processes to two or more control block combined using the link information, and the control-block processing program and in which computer reading is possible.

[0002]

[Description of the Prior Art] In order for the aftereffect of a highly informative society to roll in even to household electrical products industry and to accept want of advanced features of the home electronics in the future, it is becoming pressing need to fix base software for these home electronics or that real-time nature develops a high operating system (an operating system is written as OS.). Moreover, OS asked for real-time nature is called real-time system.

[0003] Such real-time system is asked for base control of systems, such as execution sequence management of two or more tasks, and allocation sequence management of free memory block, like OS for general purpose computers. Two or more task execution sequence management is base control that define in what kind of sequence each task is performed based on the order of time of day which changed to the height and the activation waiting state of a priority of each task, and it carries out sequential starting according to the execution sequence here, when two or more tasks are loaded to memory.

[0004] Allocation sequence management of two or more free memory block is base control of assigning any they being to a demand attaching task among two or more free memory block according to the memory size which defines in which sequence free memory block equivalent to the free area on memory being assigned to a task, and a task requires. Moreover, with real-time system, since it is notified by interruption processing, although the change of state in a household electric-appliances device etc. guarantees the real-time nature to interruption processing, it must perform execution sequence management of two or more tasks, and allocation sequence management of free memory block.

[0005] Hereafter, the conventional OS is explained. Drawing 14 is a block diagram for explaining the basic function of OS, it is what showed functional block related to execution sequence management of a task, and allocation sequence management of free memory block directly, and other parts are omitted. The interruption processing section 112 receives interruption from the input/output equipment which is not illustrated, and according to it, if required, it will call the system call service 141 which OS offers.

[0006] System call service is fundamental routine processing of the synchronours control offered from OS, exclusive control, the communication facility between tasks, a memory management function, etc. The called system call service 141 performs processing which corresponds in the system call processing section 201, and if required at this time, it will change the status of a task. When the task status is changed in the system call processing section 201, the event in which status modification of a task occurred by interruption processing is notified to a scheduler support task 202. The task status is a status flag which shows the condition that the task is waiting for I/O etc., that this status changes changes the task of an event waiting state to ready condition, or it means that the task currently performed changes to an event waiting state. In addition, there are things, such as "rel.wait", "set_flg", "wait_flg", and "rel.blk", in the system call about task management, and the system call about free memory block.

[0007] System call set_flg() is a system call which notifies having been set as the value with an event flag. System call rel_wait() is a system call which carries out forced release of the event formation waiting state of a certain task. System call wait_flg() is a system call which newly enters considering the task of arbitration as an event formation waiting state.

[0008] System call rel_blk is a system call of a purport which opens free memory block of the size of arbitration. Reflecting the task status change by interruption, a scheduler support task 202 newly determines the execution sequence of the whole task, and notifies a switch of a task to a dispatcher 203.

[0009] A dispatcher 203 interrupts a current activation task and changes the task determined by the scheduler support task 202 to a new activation task. In the real-time system as which real-time nature is required, the priority according to an urgency is attached to the task, and based on this priority, a scheduler support task 202 performs scheduling so that it may perform sequentially from a task with a high priority.

[0010] Usually, OS of dedication is not necessarily prepared to each interruption processing or a task, and OS serves as a common routine of interruption processing and a task from economization of activity memory space etc. This means that the same system call may be called, before a system call is called and processing is completed. A routine with such a property is called a reentrant routine (routine which has reentrant nature).

[0011] Drawing 15 is drawing for explaining a reentrant routine. Drawing 15 (a) is the example of the system call which is not reentrant, and drawing 15 (b) and (c) are the examples of a reentrant routine. Since drawing 15 (a) does not make it into a reentrant, it has produced [the time lag] between interruption generating and interruption processing initiation by considering system call processing as the prohibition on all interruption. An external interrupt occurs during system call processing of Task A, and drawing 15 (b) shows the example to which the call of the same system call is carried out from interruption processing. Drawing 15 (c) is an example to which the call of the system call which the external interrupt occurred while the system call of Task A was processed, the switch of Task B occurred by other system calls called from the interruption processing section, and was interrupted by interruption from the task B which switched is carried out.

[0012] As for execution sequence management of two or more tasks, or allocation sequence management of free memory block, it is ideal to be carried out by the system call with the property of such a reentrant. Moreover, execution sequence management of two or more tasks and allocation sequence management of free memory block are common at the point managed by using link structure data. Then, an example of the link structure data used for execution sequence management of two or more tasks is explained.

[0013] Drawing 16 is drawing showing an example of the link structure data which serve as the link header 90 from two or more Task Control Block (TCB). The link header 90 is also called an event flag management block, and has the link head data-address storage region 92 where the address of TCB which should be performed to the beginning among two or more TCBs is set up, and the event flag value 93 to which the value which shows the event materialized in real-time system is set.

[0014] after three tasks (a task 1, a task 2, task 3) in an activation waiting state boil three TCBs91, respectively, they correspond and perform the task corresponding to self, they have the link information 94 which shows TCB of the task which should be performed next, and the wait-condition discharge information 95 which shows conditions to cancel the activation waiting state of this task. In an example of drawing 16, if the link relation of the link header 90-link information 94 is referred to, the link information 94 of TCB1 is set up with TCB2, and the link information 94 of TCB3 and TCB3 is set up for the link information 94 of TCB2 with "termination."

[0015] It turns out that the execution sequence is given to the order [data / of drawing 16 / link structure] TCB1 →TCB2 →TCB3 →" termination." This execution sequence is defined based on the order of time of day which changed to the height and the activation waiting state of the priority about each task. Moreover, since the wait-condition discharge information 95 in TCB1 is set up with "1" and the wait-condition discharge information 95 in TCB2 is set up with "2", it turns out that it is waiting to set TCB1 and TCB2 as the value of "1" and "2" in the event flag value 93, and any TCB is still in an activation waiting state. If the event flag value 93 is set to "1" here, the event formation waiting state of TCB1 will be canceled, and if the event flag value 93 is set to "2",

the event formation waiting state of TCB2 will be canceled. Such link structure data are arranged to an external reference field. it is called an external reference field also with a global field here — having — both a system call interruption processing and a task — although — the memory area which can be accessed in common is said. A system call, interruption processing, and the memory area where each of a task can access dedication are called local field to an external reference field.

[0016] In addition, link structure data are not necessarily only the DS connected in the link information as shown above. For example, variable-length data are arranged continuously and the DS in which degree data to arbitration data can be found using the size information on each data is also link structure data. That is, when the processing-object data of arbitration can be found, the DS in which the following processing-object data can be found is called link structure data.

[0017] Then, the procedure at the time of carrying out sequential execution of two or more tasks using the link structure data shown in drawing 16 is explained. here — it should mind — it is the point of specifying the task which the system call processing section 201 should establish a processing-object storage region in a local field a processing-object storage region and next time this time, and should start this time, and the task which should be started next time. Drawing 17 is drawing showing signs (a condition (a-1), a condition (a-2), a condition (a-3), condition (a-4) condition (a-7)) that a processing-object storage region is updated this time a processing-object storage region and next time based on the link structure data shown in drawing 16, and drawing 18 is drawing showing signs that sequential execution of two or more tasks shown in link structure data is carried out. these drawings — setting — a common sign (a-1) (a-2) (a-3) shall mean a common condition

[0018] (Condition (a-1)) The system call processing section 201 sets the address of TCB shown in the link head data-address storage region 92 in the link header 90 as a processing-object storage region this time. With an example of drawing 17, since the link head data-address storage region 92 of the link header 90 is set up with TCB1, a processing-object storage region is set up with TCB1 this time. Since the task which serves as a processing object by this this time was determined as the task 1, the task which should be continuously performed to the degree of a task 1 is determined. The link information 94 of TCB1 corresponding to a task 1 is set up with TCB2 here, and the task 2 corresponding to this TCB2 must be processed next time. Then, TCB2 is set as a processing-object storage region next time, and processing is finished.

[0019] (Condition (a-2)) With reference to as what kind of event the wait-condition discharge information 95 on TCB1 which became a processing object this time is set, it determines [of which the event formation waiting state of a task 1 is canceled / or or] whether to carry out maintenance. Since the wait-condition discharge information 95 on TCB1 is set up with "1" here and the event flag value 93 of the link header 90 is set up with "1", it turns out that the event for which the task 1 was waiting was materialized. Since the event for which it was waiting was materialized, while performing a task 1, TCB1 is canceled of link structure data. In this discharge event, the link information 94 in TCB1 serves as an indeterminate. Moreover, since TCB1 was canceled of link structure data, the link head data-address storage region 92 in the link header 90 is set as TCB2.

[0020] If TCB1 is canceled of link structure data before setting a processing object as a processing-object storage region temporarily next time next time, the set point of a processing-object storage region will serve as an indeterminate next time. In the semantics which avoids this, before canceling TCB1 of link structure data, the processing-object storage region is set up next time.

(Condition (a-3)) The system call courtesy counter 201 sets the address of TCB shown in the processing-object storage region in the condition (a-2) next time as a processing-object storage region this time. Since the address of TCB2 is set as the processing-object storage region in the state of drawing 17 (a-1) next time, a processing-object storage region is set up with TCB2 this time. Since the task which serves as a processing object by this this time was determined as the task 2, the task which should be continuously performed to the degree of a task 2 is determined. The link information 94 of TCB2 corresponding to a task 2 is set up with TCB3 here, and the task 3 corresponding to this TCB3 must be processed next time. Then, TCB3 is set as a processing-object storage region next time, and processing is finished.

[0021] (Condition (a-4)) With reference to as what kind of event the wait-condition discharge information 95 on TCB2 which became a processing object this time is set, it determines whether to cancel the event formation waiting state of a task 2, or maintain. Since the wait-condition discharge information 95 on TCB2 is set up with "2" here and the event flag value 93 of the link header 90 is set up with "1", it turns out that the event for which the task 2 was waiting is abortive. Since it is event failure, activation of a task 2 is skipped. With an activation skip, setting out of the link head data-address storage region 92 in the link header 90 serves as as [TCB2].

[0022] (Condition (a-5)) The system call courtesy counter 201 sets the address of TCB shown in the processing-object storage region in the condition (a-3) next time as a processing-object storage region this time. Since the address of TCB3 is set as the processing-object storage region in the state of drawing 17 (a-3) next time, a processing-object storage region is set up with TCB3 this time. Since the task which serves as a processing object by this time was determined as the task 3, the task which should be continuously performed to the degree of a task 3 is determined. The link information 94 of TCB3 corresponding to a task 3 is set up with termination here, it is next time and the task activation in link structure data must be ended. Then, termination is set as a processing-object storage region next time, and processing is finished.

[0023] (Condition (a-6)) With reference to as what kind of event the wait-condition discharge information 95 on TCB3 which became a processing object this time is set, it determines whether to cancel the event formation waiting state of a task 3, or maintain. Since the wait-condition discharge information 95 on TCB3 is set up with "1" here and the event flag value 93 of the link header 90 is set up with "1", it turns out that the event for which the task 3 was waiting was materialized. Since it is condition coincidence, a task 3 is performed and TCB3 is canceled of link structure data. In this discharge event, the link information 94 in TCB3 serves as an indeterminate. The link head data-address storage region 92 in the link header 90 shifts to a condition (a-7), set it as TCB2, and ends processing.

[0024] When performing each task as mentioned above, in a processing-object storage region and the condition that a processing-object storage region and the link head data-address storage region 92 of the link header 90 are unfixed next time, it is not desirable for interruption processing to occur this time. That is, it turns out that interruption processing should be forbidden this time in a processing-object storage region, the condition (a-1) that a processing-object storage region and the link head data-address storage region 92 of the link header 90 become unfixed next time and a condition (a-2), the condition (a-3), the condition (a-4), the condition (a-5), and the condition (a-6), and a permission should be granted in the section of ***** which these have decided.

[0025] Drawing 19 is drawing in which the authorization section of interruption processing generating and the prohibition section were established by the above viewpoint. Moreover, drawing 20 is drawing showing assignment of the memory area at the time of interruption processing generating. In drawing 20, the link header 90 and link structure data are arranged to the external reference field, processing of the interrupted side and processing of the side which interrupted are calling system call service uniquely, respectively, and these have secured the local field, respectively. In the secured local field, that the processing-object storage region is secured next time turns out to be a processing-object storage region this time.

[0026] Then, in order to perform execution sequence management of two or more tasks, and allocation sequence management of free memory block, guaranteeing the real-time nature to interruption processing, it explains what kind of point must be cared about. When permitting interruption processing, another thing must be cared about in execution sequence management of two or more tasks. It is that a system call may be performed also in that interruption processing, and link structure data may be changed by this system call, when interruption processing occurs in the interruption processing authorization section. If link structure data are changed in interruption processing, a processing-object storage region and the link structure data changed [in / next time / a processing-object storage region and interruption processing] will become inharmonious, and will cause malfunction of real-time system this time in the local field of the interrupted side.

[0027] Henceforth, it explains how the inequality of a processing-object storage region and the link structure data changed by the time of interruption processing arises this time processing-object storage region in a local field, and next time, the system call of a purport which interruption processing generates drawing 21 in the interruption processing authorization section, and sets an

event flag as "2" in this interruption processing — the case where "set_flg(2)" is called — this time — a processing-object storage region — It is drawing showing how a processing-object storage region changes next time. Drawing 22 When interruption processing occurs in the interruption processing authorization section and system call "set_flg(2)" is called in this interruption processing, it is drawing showing how link structure data change.

[0028] Although the content of the processing-object storage region is the same as that of drawing 17 and drawing 18 in drawing 21 and drawing 22 a processing-object storage region and next time the link structure data in a condition (a-1) – a condition (a-4), and this time Since interruption processing occurred in the condition of ** in drawing 17 , a condition (b-1) – a condition (b-5) occur after a condition (a-2). According to generating of this condition (b-1) – a condition (b-5) Processing of processing of the interrupted side is replaced with what is shown in a condition (a-8) – a condition (a-9) from what is shown in a condition (a-5) – a condition (a-7).

[0029] These conditions are explained below.

(Condition (b-1)) Since interruption processing occurred, the system call processing section 201 secures a local field for interruption processings, and prepares a processing-object storage region here a processing-object storage region and next time this time. Since set_flg(2) was called in this interruption processing, the event flag value 93 is updated from "1" to "2." Moreover, since the address of TCB2 is set as the link head data-address storage region 92 in the condition (a-1) of drawing 21 , a processing-object storage region is set up with TCB2 this time in the newly secured local field. The link information 94 of TCB2 corresponding to a task 2 is set up with TCB3, and must process the task 3 corresponding to this TCB3 next time. Then, TCB3 is set as a processing-object storage region next time, and processing is finished.

[0030] (Condition (b-2)) With reference to as what kind of event the wait-condition discharge information 95 on TCB2 which became a processing object this time is set, it determines whether to cancel the event formation waiting state of a task 2, or maintain. Since the wait-condition discharge information 95 on TCB2 is set up with "2" here and the event flag value 93 of the link header 90 is set up with "2", it turns out that the event for which the task 2 was waiting is materialized. Since it is condition coincidence, after performing a task 2, TCB2 is canceled of link structure data. Then, the link head data-address storage region 92 in the link header 90 is set as the address of TCB3.

[0031] (Condition (b-3)) The system call courtesy counter 201 sets the address of TCB shown in the processing-object storage region in the condition (b-2) next time as a processing-object storage region this time. In the condition (b-2), since the address of TCB3 is set as the processing-object storage region next time, a processing-object storage region is set up with TCB3 this time. Since the task which serves as a processing object by this this time was determined as the task 3, the task which should be continuously performed to the degree of a task 3 is determined. The link information 94 of TCB3 corresponding to a task 3 is set up with termination here, it is next time and the task activation in link structure data must be ended. Then, termination is set as a processing-object storage region next time, and processing is finished.

[0032] (Condition (b-4)) With reference to as what kind of event the wait-condition discharge information 95 on TCB3 which became a processing object this time is set, it determines whether to cancel the event formation waiting state of a task 3, or maintain. Since the wait-condition discharge information 95 on TCB3 is set up with "1" here and the event flag value 93 of the link header 90 is set up with "2", it turns out that the event for which the task 3 was waiting is abortive. Since it is condition failure, a task 3 is not performed and TCB3 is not canceled of link structure data. Thereby, the link head data-address storage region 92 in the link header 90 ends processing in a condition (b-5), while it had been set as TCB3 by it.

[0033] (Condition (a-8)) Since interruption processing was ended in the condition (b-5), it shifts to the condition (condition (a-8)) immediately after the condition (a-2) of processing of the interrupted side. In the system call courtesy counter 201 in processing of the side interrupted here, the content to which the processing-object storage region was set in the condition (a-1) is saved as it is a processing-object storage region and next time this time. That is, the processing-object storage region is set as TCB1-TCB2 a processing-object storage region and next time this time, respectively.

[0034] (Condition (a-9)) The system call courtesy counter 201 of processing of the interrupted

side tends to set up a processing object a processing object and next time this time a processing-object storage region and next time according to the content of setting out with a processing-object storage region. By copying the value of a processing-object storage region to a processing-object storage region here this time next time, a processing-object storage region is set as the address of TCB2 this time. Then, in processing of the side which interrupted, although it is going to set up the value of a processing-object storage region next time, since TCB2 is canceled of link structure data, TCB of the task which should be performed next cannot be set to a processing-object storage region next time. Therefore, the content in a processing-object storage region serves as an indeterminate value next time.

[0035] Since TCB2 is canceled of link structure data and the task 2 is excepted from memory, if it is going to perform the task 2 corresponding to TCB2, a system will lapse into processing impossible and malfunction of a system will already generate it. If link structure data are changed in interruption processing the link head data-address storage region 92 and this time as mentioned above even if it is the definite section of a processing-object storage region a processing-object storage region and next time, there is a possibility that the inequality of a processing-object storage region and the link structure data changed by the time of interruption processing may occur, and there is fear of malfunction a processing-object storage region and next time this time.

[0036] Since link structure will be changed by processing of another side even if it holds the information about a link in one processing when making reentrant processing the system call treating link structure data, this means that it becomes impossible to maintain the consistency of the information about the link currently held. This problem arises similarly, not only when the same system call is called according to interruption generating in a reentrant system call but even when other system calls which change the link structure data which were being treated during reentrant system call processing interrupt and it is called according to generating. Therefore, in the conventional OS, in order not to change link structure during processing of link structure data, the interrupt control method of forbidding external interruption was used until processing of all data connected to the link was completed.

[0037] However, by this interrupt control method, there is a problem that interrupt, so that the number of data connected to the link increases, and prohibition time amount becomes long, and the time lag generated between actual interruption generating and interruption processing initiation as shown in drawing 15 (a) becomes large. That interrupt in this system and prohibition time amount is long since real-time system has semantics with the important real-time nature to external interruption can become the cause of giving a fatal defect to a system. picking of commo data according to external interruption directly -- spilling -- etc. -- there is a problem. There is a problem that the time amount to which a system does not operate increases as a priority by delay of the task switch by interruption having been delayed indirectly etc.

[0038] Therefore, the application which uses OS needs to perform a system design in consideration of the interruption prohibition time amount in the case of processing the maximum number connected with link structure data. Moreover, depending on the case, the system construction in the maximum interruption prohibition time amount becomes impossible, and a solution, such as having the hardware of dedication, is needed. In order to solve the problem of the increment in the interruption prohibition time amount at the time of link structure data processing of OS, with the technique shown in "JP,9-311794A", it has proposed forming the flag for corresponding to a link header and notifying modification of link structure data. That is, this conventional technique sets the value which shows modification as a flag, when the system call processing section of the side which interrupted changes link structure data. If it detects that were supervising change of a flag in the interruption authorization section, and the flag was set as the value a system call indicates modification to be, the system call processing section of the interrupted side is redoing processing from the initial data of a link, and even if link data are changed, it will guarantee that a right result is obtained. If this conventional technique is used, it will become possible to shorten interruption prohibition time amount.

[0039] Interruption processing occurs in the interruption processing authorization section, it sets to this interruption processing, and drawing 23 is system call service. When set flag (2) is called, this time Processing-object storage region. It is drawing showing how a processing-object storage region changes next time. Drawing 24 Interruption processing occurs in the interruption processing

authorization section, and it sets to this interruption processing, and is system call service. When set_flg (2) is called, link structure data and flag information are drawings showing how it changes. these drawings — setting — a common sign (a-1) (a-2) (a-3) shall mean a common condition [0040] That drawing 24 differs from drawing 22 is the point that flag information is established in the link structure data of drawing 24 , and the flag information concerned is set as "1" in the condition (b-2). In this condition (b-2), since TCB2 is canceled of link structure data and the DS of link structure data changed by this discharge, flag information was changed into "1" from "0" in the semantics which shows this modification.

[0041] That drawing 23 differs from drawing 21 is the point that subsequent processings are transposed to the condition (a-10) and the condition (a-11) from the condition (a-9), when a return is carried out to processing of the side interrupted from processing of the side which interrupted. (Condition (a-8)) Since processing was ended in the condition (b-5), it shifts to the condition (condition (a-8)) immediately after the condition (a-2) of processing of the interrupted side. In the system call courtesy counter 201 in processing of the side interrupted here, the content to which the processing-object storage region was set in the condition (a-1) is saved as it is a processing-object storage region and next time this time. That is, the processing-object storage region is set as TCB1-TCB2 a processing-object storage region and next time this time, respectively.

[0042] (Condition (a-10)) The system call courtesy counter 201 judges whether in processing of the side interrupted with reference to the flag in link structure data, modification arose to link structure data. The flag of link structure data is set as "1" which shows the purport which had modification in the DS of link structure data here.

[0043] (Condition (a-11)) Since the flag of link structure data shows the purport with modification, by copying the content of the link head data-address storage region 92 in the link header 90 to a processing-object storage region this time, the system call courtesy counter 201 of processing of the interrupted side sets a processing-object storage region as TCB3 this time, and sets the address shown in the link information 94 of TCB3 as a processing-object storage region next time. Since TCB3 exists in link structure data actually and the task 3 exists also in memory, this task 3 will be performed normally.

[0044]

[Problem(s) to be Solved by the Invention] However, with this conventional technique, in the system call of the side interrupted, whenever it processes one TCB, the monitor of a flag is required, and only in "the number of time amount x data which flag monitor processing takes", the time amount required in order to decide a processing object processing-object-next time this time becomes long. If it says with an example of drawing 19 , in all the authorization sections equivalent to **, **, **, and **, it will have to supervise whether flag information was updated and great time amount will be spent on this flag monitor. That is, in the advanced technology shown in "JP,9-311794,A", although the real-time nature through which it interrupts and passes improved, when an interrupt does not occur, the processing delay based on the increment in unnecessary flag monitor time amount will become remarkable.

[0045] If the time amount which decision of a processing object takes processing-object-next time this time becomes long, the system operation time which can be assigned to the task corresponding to TCB will become short. If the system operation time assigned to the task becomes short, constraint of as opposed to an application program only in the part will become severe, and the design of real-time system will become difficult.

[0046] The object of this invention is offering the record medium which recorded the control-block processor and control-block processing program which can shorten the prohibition time amount of interruption processing and in which computer reading is possible, without needing the monitor of the flag set point.

[0047]

[Means for Solving the Problem] The link structure data which consist of two or more control block which a link information is used for this invention and combined in order to attain the above-mentioned object, The address storage region where the address of which control block is written in among two or more control block, The actuation which determines control block by which the address is written in the address storage region as a processing object this time, Control block of the link place of the control block is determined as a processing object next time. The processing-

object decision means which repeats the actuation which writes the address of a processing object in an address storage region the next time concerned about all control block in link structure data. A processing means to perform processing about a processing object this time [this] whenever a processing object is determined this time. It has an authorization means to permit generating of interruption processing in a period after processing about a processing object is performed this time until a processing object is determined as a degree this time. Said processing-object decision means While it is started when interruption processing occurs in the period when interruption processing was permitted, and determining which control block in link structure data as a processing object this time in interruption processing It determines as a processing object the next time which should be processed in case processing of the side interrupted in another control block is resumed. [when resuming processing of the side which it has the reentrant routine which writes the address of a processing object in an address storage region the next time concerned, and was interrupted after interruption processing generating, and when interruption processing does not occur] It is characterized by determining control block by which the address is written in the address storage region as a processing object this time.

[0048]

[Embodiment of the Invention] Hereafter, the operation gestalt of a control-block processor is explained, referring to a drawing. Here, the gestalt of a negotiation and dealing of the control-block processor commonly used by this contractor in the actual condition is a gestalt which records the software of the execute form of the operating system which has a function equivalent to a control-block processor on a record medium, and is circulated and dealt in as a software package. When such a software package is installed in the device which has a processor and memory by the customer who purchased this and the installed device performs processing as this software, a device achieves the function as a control-block processor. It is more appropriate to consider the software which performs the content of processing as a control-block processor using these hardware resources rather than considering that the above gestalten are used commonly the functional subject as a control-block processor calls it hardware resources, such as a processor and memory. Moreover, since it is common to consist of two or more subroutines and work areas as for the software which has the complicated content of processing, it should be considered to be the component with which each subroutine and work area became independent. However, there are also many things which the existing operating system has library-ized in these subroutines and work areas, and explanation which made it the point of argument whether for them to have used as a component the subroutine which achieves what kind of function in order to avoid the explanation and to achieve the function of a control-block processor, and a work area, and to have developed them since it is most which does not deserve explanation is given.

[0049] (The 1st operation gestalt) The 1st operation gestalt is explained hereafter, referring to a drawing. Drawing 1 is the block diagram having shown an example of a system which carried the operating system (it is henceforth described as OS). The arrow head of the continuous line in a block diagram shows transition of processing, and the arrow head of a broken line shows transition of data (the arrow head in a block diagram is hereafter made into the same semantics). This system consists of the program control equipment 101 by CPU, an input unit 102, an output unit 103, and a memory apparatus 104. Furthermore, program control equipment 101 consists of OS's 114 which control the execution sequence of the interruption processing section 112 which receives an input from the input/output control 111 which controls I/O with an input unit 102 and an output unit 103, and input/output control, the task 113 which divided the function of a program into some, and a task 113.

[0050] Program control equipment 101 performs program execution according to the processing program memorized by the memory apparatus 104. OS 114 is equivalent to a control-block processor, and controls the execution sequence of a task 113 by switching an activation task based on the priority and the task status which were added to the task 113. A priority and the task status are changed when the interruption processing section 112 and a task 113 call the system call service 141 which OS 114 offers.

[0051] Input/output control 111 notifies the incoming event received from the input unit 102 to the interruption processing section 112 as external interruption. Even if, as for the carrier beam interruption processing section 112, other control sections are performing advice, it performs

promptly. There are two or more functions, such as a synchronous control, exclusive control, communication facility between tasks, and a memory management function, in the system call service 141.

[0052] Drawing 2 is the link header 90 arranged to the external reference field of a memory apparatus 104, and drawing showing an example of link structure data. after the task in an activation waiting state boils TCB91 in this Fig., respectively, it corresponds and performs the task corresponding to self, it has the link information 94 which shows TCB of the task which should be performed next, and the wait-condition discharge information 95 which shows conditions to cancel the activation waiting state of this task.

[0053] The link header 90 is also called an event flag management block, and has the link head data-address storage region 92 where the address of TCB which should be performed to the beginning among two or more TCBs is set up, and the event flag value 93 to which the value which shows the event materialized in real-time system is set. In these points of providing, although the link header 90 is not different from the thing of the conventional technique, it differs from the conventional technique in that the processing-object data-address storage region 105 is added to the link header 90 next time. That is, the processing-object data-address storage region 105 (it is equivalent to a processing-object storage region the next time arranged to the local field in the conventional technique.) is arranged with this operation gestalt next time at the link header 90 in an external reference possible field.

[0054] Drawing 3 is the block diagram showing functionally the configuration of the system call processing section 201 in processing of the side interrupted by interruption processing. As shown in drawing 3, the system call service 141 in the 1st operation gestalt serves as the Main control section 301 and the interrupt control section 302 from the processing-object data-address storage region 310 the processing-object data decision section 305, the data-processing section 306, and this time next time this time with the processing-object data-address acquisition section 303 and the data-processing termination judging section 304.

[0055] The Main control section 301 performs a series of processings from initiation of system call processing to termination using each control section. The interrupt control section 302 forbids or permits registration of external interruption. From the link header installed in the field in which external reference is possible, the processing-object data-address acquisition section 303 extracts the address information stored in the processing-object data-address storage region 105 a link head data-address storage region or next time, and stores it in the processing-object data-address storage region 310 this time.

[0056] The data-processing termination judging section 304 determines [which continues processing of link structure data / or] whether to carry out termination from the information on the processing-object data-address storage region 310 this time. The processing-object data decision section 305 draws a processing object from a processing object next time this time, and sets up the address of the data for which the processing-object data-address storage region 105 was asked the next time which exists in the link header 90 next time.

[0057] The data-processing section 306 performs a certain processing to the task which is equivalent to a processing object this time. Processing of the system call service 141 at the time of interruption is further explained to a detail using the block diagram of drawing 3, and the flow chart which shows drawing 4. Drawing 4 is a flow chart which shows the content of processing of the system call service 141 at the time of interruption.

[0058] If processing of link structure data is first started in the system call processing section, in step S402, registration of interruption will be forbidden in the interrupt control section 302. Next, at step S403, the processing-object data-address acquisition section 303 extracts the head data address of a link from the link head data-address storage region of a link header, and stores in the processing-object data-address storage region 310 this time. A processing-object data address serves as information which shows the termination of a link by the way this time.

[0059] Then, at step S404, a processing-object data address investigates whether it is the termination information on a link after setting up a processing-object data address this time when the data-processing termination judging section 304 was set up this time. If it is termination, processing of link structure data will be ended (step S404- Yes), and processing will be continued if it is not termination (step S404- No). When processing is continued, in step S405, the processing-

object data decision section 305 asks for the address of processing-object data from the internal information of processing-object data next time this time, and stores in the processing-object data-address storage region 105 next time next time. Since the link structure data of this operation gestalt constitute the link from a link information in data, they store the content of this link information in the processing-object data-address storage region 105 next time.

[0060] At step S406 next, the data-processing section 306 performs data processing about the task which is equivalent to processing-object data this time. After data processing is completed, in step S407, in the interrupt control section 302, the receptionist condition of interruption is temporarily changed into an authorized state from a prohibition condition, and registration of interruption processing is returned to a prohibition condition in step S408 after that.

[0061] Then, at step S409, the processing-object data-address acquisition section 303 extracts a processing-object data address the next time stored in the processing-object data-address storage region 105 next time, and stores in the processing-object data-address storage region 310 this time. Henceforth, processing of step S404 – step S409 is repeated until it detects termination of processing in the data-processing termination judging section 304. In addition, processing of link structure data may be ended by fulfilling the conditions of arbitration in step S406 besides discovering the termination of link data.

[0062] Drawing 5 is the block diagram showing the function of the system call processing section 201 which performs interruption processing. As shown in drawing 5, the system call processing section 201 which performs processing of the side which interrupted turns into the Maine control section 501, the data-processing section 502, and the link structure data modification judging section 503 from the processing-object data-address initialization section 504 next time.

[0063] The Maine control section 501 performs a series of processings from initiation of the system call processing called from interruption processing to termination using each control section. The data-processing section 502 performs a certain processing to which task equivalent to TCB in link structure data. The link structure data modification judging section 503 judges whether link structure was changed in the data-processing section 502.

[0064] The processing-object data-address initialization section 504 sets the content of the link head data-address storage region 92 as the processing-object data-address storage region 105 next time next time, when it is judged that link structure was changed in the link structure data modification judging section 503. It explains to a detail using the flow chart which shows processing of the system call processing section 201 constituted like drawing 5 to drawing 6. In addition, the flow chart of drawing 6 adds the flow chart of other system call processings in which it interrupted during system call processing to the flow chart of the processing of the interrupted side shown in drawing 4.

[0065] In the flow chart of drawing 4, data processing is completed in the system call of arbitration between step S407 and step S408, and registration of interruption is permitted temporarily. A system call is called out of the interruption processing to which the external interrupt occurred and was called in this authorization section, and processing of a system call is started.

[0066] Initiation of processing of the side which interrupted performs processing about the task by which the data-processing section 502 is equivalent to a processing object in step S602 this time. In data processing, add TCB to the arbitrary locations of a link, TCB in a link is canceled, or processing unrelated to link structure data is performed. After performing processing about a task, processing will be continued, if it is judged whether the link information of link structure data was changed by the link structure data modification judging section 503 in step S603, the link information is not changed, system call processing is ended (nothing [step S603-modification]) and the link information is changed (step S603-*****).

[0067] When the link information is changed in step S604, by the processing-object data-address initialization section 504, the link head data address of the link structure data after modification is extracted from the link head data-address storage region of the link header 90, and it stores in the processing-object data-address storage region 105 next time next time. Then, it explains, referring to drawing 9 about near processing in which the system call processing section 201 interrupted while referring to the example of link structure data, and processing of the interrupted side.

Drawing 9 is a signs [that the processing-object data-address storage region 105 is updated next time] condition (c-1), a condition (c-2), a condition (c-3), and a condition (c-4)..... It is drawing

showing a condition (c-7). the system call of the purport by which this example sets an event flag as "1" — the case where "set_flg (1)" is called is assumed.

[0068] (Condition (c-1)) System call Since set_flg (1) began to call and come out, the system call service 141 sets value "1" as the event flag value 93 in the link header 90. On the other hand, as shown in drawing 9, TCB1-TCB2-TCB3-TCB4 is arranged at link structure data, and TCB1 is set to the link header 90. Moreover, "1" is set to the event flag value 93. Furthermore, TCB1 is set to the processing-object data-address storage region 105 next time. TCB1, TCB2, TCB3, and TCB4 are waiting to set an event flag as "2", "2", "1", and "3", respectively. In this condition, set_flg (1) should occur and the event flag should be set as "1." set_flg (1) is a system call which notifies that the event flag was set as "1" here. In this case, it is necessary to cancel TCB3 which was waiting to set an event flag as "1" of an event formation waiting state. By step S401 – step S402, TCB1 serves as a processing object first this time. Thus, when TCB1 is set as a processing object this time, link structure data come to be shown in a condition (c-1).

[0069] (Condition (c-2)) Since a processing object is not termination this time, step S404 serves as No and shifts to step S405. At step S405, TCB2 set as the link information 94 of TCB1 is copied to the processing-object data-address storage region 105 next time, as shown in the arrow head of a broken line. Thereby, a processing object serves as TCB2 next time. After setting up a processing object next time, processing of a task 1 is performed in step S406. If the wait-condition discharge information 95 on the task which is going to carry out also here and it is going to perform the value of an event flag and after this is in agreement, the event formation waiting state of the task concerned will be canceled. However, the value of the wait-condition discharge information 95 on a task 1 is "2", since the value of an event flag is "1", it becomes inharmonious, and TCB1 remains to link structure data, without being canceled of link structure data. After skipping activation of a task 1, it shifts to step S407. Interruption processing is permitted at step S407. Thus, interruption processing should occur immediately after [at the time of permitting interruption processing].

[0070] (Condition (c-3)) At the time of generating of this interruption processing, it shifts to processing of step S601. System call rel_wait (task2) should be called here at the time of interruption processing activation of step S602. rel_wait (task2) is a system call which carries out forced release of the event formation waiting state of TCB2 here. TCB2 is canceled of link structure data by activation of this rel_wait (task2), and shifts to step S603 by it. Since TCB2 was canceled of link structure data, it means that modification had arisen to link structure data, and step S603 serves as Yes. If step S603 serves as Yes, TCB1 shown in the link head data-address storage region 92 in step S604 will be set as the processing-object data-address storage region 105 next time. At this event, the processing-object data-address storage region 105 is set up with TCB1 next time.

[0071] (Condition (c-4)) Interruption processing is completed by the above processing. Processing of system call set_flg (1) is resumed by termination of interruption processing. Since it is step S407 which interruption processing generated, processing of set_flg (1) is resumed from step S408. Since TCB1 was set as the processing object here next time by step S408 in interruption processing – step S409, processing of link structure data is redone from TCB1. That is, step S404 – step S409 are repeatedly performed until step S404 is set to Yes.

[0072] The repeat of step S404 – step S409 shows link structure data to a condition (c-4) eventually. Then, it explains, referring to drawing 10 about the example of processing when new TCB is inserted in link structure data in processing of the interrupted side.

(Condition (d-1)) As shown in drawing 10, TCB2-TCB3-TCB4 is arranged at link structure data, and TCB1 is set to the link header 90. here — system call set_flg (1) is call appearance — the event flag value [in / in the system call service 141 / the link header 90] 93 since it was carried out — a value — "1" is set up. Moreover, TCB2 is set to the processing-object data-address storage region 105 next time. The link information 94 between TCB2, TCB3, and TCB4 shall be set up in order of the priority of each task. I make a priority into an equal at the number of each task, and think that a priority is as high as a thing with few this task number here.

[0073] Moreover, these tasks are waiting to set an event flag as "2", "1", and "3", respectively. In this condition, set_flg (1) should occur and the event flag should be set as "1." In this case, it is necessary to cancel TCB3 which was waiting to set an event flag as "1" of an event formation

waiting state. By step S401 – step S402, TCB2 serves as a processing object first this time. When TCB2 is set as a processing object this time, link structure data come to be shown in a condition (d-1).

[0074] (Condition (d-2)) Since a processing object is not termination this time, step S404 serves as No and shifts to step S405. At step S405, TCB3 set as the link information 94 of TCB2 is copied to the processing-object data-address storage region 105 next time. Thereby, a processing object serves as TCB3 next time. After setting up a processing object next time, processing of a task 2 is performed in step S406. If the wait-condition discharge information 95 on the task which is going to carry out also here and it is going to perform the value of an event flag and after this is in agreement, the event formation waiting state of the task concerned will be canceled. However, the value of the wait-condition discharge information 95 on a task 2 is “2”, and since the value of an event flag is “1”, it becomes inharmonious, and TCB2 remains to link structure data, without being canceled of link structure data. After skipping activation of a task 2, it shifts to step S407. Interruption processing is permitted at step S407. Thus, interruption processing should occur immediately after [at the time of permitting interruption processing].

[0075] (Condition (d-3)) At the time of generating of this interruption processing, it shifts to processing of step S601. System call wait_flg (task1) should be called here at the time of interruption processing activation of step S602. wait_flg (task1) is a system call which newly adds a task 1 to link structure data as an event formation waiting state, and TCB1 is added to link structure data by activation of this wait_flg (task1), and shifts to step S603 by it. Since TCB1 was added to link structure data here, it means that modification had arisen to link structure data, and step S603 serves as Yes. If step S603 serves as Yes, TCB1 shown in the link head data-address storage region 92 in step S604 will be set as the processing-object data-address storage region 105 next time. At this event, the processing-object data-address storage region 105 is set up with TCB1 next time.

[0076] (Condition (d-4)) Interruption processing is completed by the above processing. Processing of set_flg (1) is resumed by termination of interruption processing. Since it is step S407, that interruption processing occurred resumes processing of set_flg (1) from step S408. Since TCB1 was set as the processing object here next time by step S408 in interruption processing – step S409, processing of link structure data is redone from TCB1. That is, step S404 – step S409 are repeatedly performed until step S404 is set to Yes.

[0077] The repeat of step S404 – step S409 shows link structure data to a condition (d-4) eventually. While being able to stop the increment in the system call processing time at the time of shortening interruption prohibition time amount as mentioned above to the minimum according to this operation gestalt, the time amount by which a task control is not performed as the meaning of the priority added to the task can be shortened, and it is seldom necessary to take the execution time of OS into consideration. Thereby, the efficiency of the application design in real-time system construction is increased.

[0078] (The 2nd operation gestalt) Although the processing-object storage region was set as what is shown in the link head data-address storage region 92 with the 1st operation gestalt next time, it is the operation gestalt which consisted of 2nd operation gestalten so that processing of the side which interrupted might be resumed from the next TCB of TCB currently performed at the end in processing of the interrupted side.

[0079] Drawing 7 is the block diagram showing functionally the configuration of the system call processing section 201 in processing of the side which interrupted. By the block shown in drawing 7, the block diagram of other system call processings in which it interrupted during system call processing is added to the block diagram indicated to drawing 3 for the above re-start processes. However, the part already shown in drawing 3 is omitted.

[0080] As shown in drawing 7, the system call service 141 in the 2nd operation gestalt serves as the Maine control section 701, the resumption data position-memory section 702 of processing, the data-processing section 703, the link structure data modification judging section 704, and the resumption data fixing section 705 of processing from the resumption data location storage region 710 of processing. The Maine control section 701 performs a series of processings from initiation of the system call processing called from interruption processing to termination using each control section.

[0081] The resumption data position-memory section 702 of processing is stored in the resumption data location storage region 710 of processing the next time stored in the next processing-object data-address storage region 105 of a link header by making a processing-object data address into the resumption data location of processing. With it, the priority for [this] a restart and the link place for a restart are saved in the resumption data location storage region 710 of processing.

[0082] The data-processing section 703 performs processing about the task corresponding to TCB of the arbitration in link structure data. The link structure data modification judging section 704 judges whether link structure was changed in the data-processing section 703. The resumption data fixing section 705 of processing sets up the link data address optimal for resumption of processing next time for the processing-object data-address storage region 105, when it is judged that link structure was changed in the link structure data modification judging section 704.

[0083] It explains to a detail further using the flow chart which shows processing of the system call processing section 201 concerning the 2nd operation gestalt constituted as shown in the block diagram of drawing 7 to drawing 8. In addition, the flow chart of drawing 3 adds the flow chart of other system call processings in which it interrupted during system call processing to the flow chart of near processing in which the 1st operation gestalt shown in drawing 4 interrupted.

[0084] When data processing is completed in the system call of arbitration and registration of interruption is permitted temporarily (between step S407 and step S408), an external interrupt occurs, a system call is called out of the interruption processing to which it was called, and processing of a system call is started. First, it stores in the resumption data location storage region 710 of processing by making a processing-object data address applicable to a restart the next time when the resumption data position-memory section 702 of processing is stored in the next processing-object data-address storage region 105 of a link header in step S802. With it, the priority for [this] a restart and the link place for a restart are saved in the resumption data location storage region 710 of processing.

[0085] Next, at step S803, the data-processing section 703 performs processing about processing-object data this time. In processing here, add data to the arbitrary locations of a link, TCB in link structure data is canceled of a link, or processing unrelated to link structure data is performed. After performing data processing in step S804, processing will be continued, if it judges whether data processing changed the link information of link structure data by the link structure data modification judging section 704, and the link information is not changed, system call processing is ended (nothing [804-modification]) and the link information is changed (804-*****).

[0086] When the link information is changed, at step S805, it judges whether in data processing of step S804, the object for a restart was canceled of link structure data. If that is not right, it will shift to step S807, but if it becomes so, it will shift to step S806. At step S806, in data processing of step S804, when the system call of a purport which cancels the event formation waiting state of a task is performed and the object for a restart is canceled, the link place for [which was saved] a restart is set up as a processing object next time.

[0087] At step S807, it judges whether the task was inserted in link structure data in data processing of step S804. At step S808, it judges which [of the priority of the newly inserted task for a restart] is high. If the priority for a restart is high, it will shift to step S810, but if the priority of the newly inserted task is high, it will shift to step S809.

[0088] At step S809, the newly inserted task is set up as a processing object next time. At step S810, the object for a restart is set up as a processing object next time. Then, an example of the processing in the flow chart of drawing 8 is explained, referring to the explanatory view of drawing 11. Drawing 11 is drawing showing signs that link structure data are changed by processing shown in the flow chart of drawing 8.

[0089] (Condition (c-1)) As shown in drawing 11, TCB1-TCB2-TCB3-TCB4 is arranged at link structure data, and TCB1 is set to the link header 90. Moreover, "1" is set to the event flag value 93. Moreover, TCB1 is set to the processing-object data-address storage region 105 next time. TCB1, TCB2, TCB3, and TCB4 are waiting to set an event flag as "2", "2", "1", and "3", respectively. In this condition, set_flg (1) should occur and the event flag should be set as "1." In this case, it is necessary to cancel TCB4 which was waiting to set an event flag as "1" of an event formation waiting state. By step S401 - step S402, TCB1 serves as a processing object first this time. Thus, when TCB1 is set as a processing object this time, link structure data come to be

shown in a condition (c-1).

[0090] Since a processing object is not termination this time, step S404 serves as No and shifts to step S405. At step S405, TCB2 set as the link information 94 of TCB1 is copied to the processing-object data-address storage region 105 next time. Thereby, a processing object serves as TCB2 next time. After setting up a processing object next time, in step S406, it is going to perform processing of a task 1. If the wait-condition discharge information 95 on the task which is going to carry out also here and it is going to perform the value of an event flag and after this is in agreement, the event formation waiting state of the task concerned will be canceled. However, the value of the wait-condition discharge information 95 on a task 1 is "2", and since the value of an event flag is "1", it becomes inharmonious, and TCB1 remains to link structure data, without being canceled of link structure data. After skipping activation of a task 1, it shifts to step S407. Interruption processing is permitted at step S407. Thus, interruption processing should occur immediately after [at the time of permitting interruption processing].

[0091] (Condition (c-2)) At the time of generating of this interruption processing, it shifts to processing of step S801. In step S802, TCB2 is set as the object for a restart here, and the priority for [this] a restart and TCB3 used as the link place for a restart are saved. The object for a restart is set up and interruption processing is performed in step S803 after preservation of the priority for a restart, and the link place for a restart. System call rel_wait(task2) should be called at the time of this interruption processing activation. TCB2 is canceled of link structure data by activation of this rel_wait(task2), and shifts to step S804 by it. Since TCB2 was canceled of link structure data, it means that modification had arisen to link structure data, and step S804 serves as Yes. If step S804 serves as Yes, it will shift to step S805. Since TCB2 was canceled by rel_wait(task2), TCB which serves as a processing object next time at the time of a restart will have disappeared by rel_wait(task2). Therefore, step S805 serves as Yes and shifts to step S806. Although the object for a restart was canceled here, TCB3 which was a link place for [this] a restart is saved in step S802. At step S806, as shown in an arrow head a1 next time by making into a processing object TCB3 which is this link place, it sets up.

[0092] (Condition (c-3)) Interruption processing is completed by the above processing. Processing of set_flg(1) is resumed by termination of interruption processing. Since it is step S407 which interruption processing generated, processing of set_flg(1) is resumed from step S408. Since TCB3 was set as the processing object here next time by step S408 in interruption processing - step S409, processing of link structure data is redone from TCB3. That is, step S404 - step S409 are repeatedly performed until step S404 is set to Yes.

[0093] In addition, when TCB canceled of link structure data is except [for a restart], setting out of a processing object is not performed next time. Then, an example of processing of the flow chart of drawing 12 is explained, referring to the explanatory view of drawing 12. Drawing 12 is drawing showing signs that link structure data are changed by processing shown in the flow chart of drawing 8.

[0094] (Condition (d-1)) TCB2-TCB3-TCB4 is arranged at link structure data, and TCB2 is set to the link head data-address storage region 92 of the link header 90. Moreover, "1" is set to the event flag value 93. Moreover, TCB3 is set to the processing-object data-address storage region 105 next time. TCB2, TCB3, and TCB4 are waiting to set an event flag as "2", "1", and "3", respectively. In this condition, set_flg(1) should occur and the event flag should be set as (1). In this case, it is necessary to cancel TCB3 which was waiting to set an event flag as "1" of an event formation waiting state. By step S401 - step S402, TCB1 serves as a processing object first this time. Thus, when TCB1 is set as a processing object this time, link structure data come to be shown in a condition (d-1).

[0095] Since a processing object is not termination this time, step S404 serves as No and shifts to step S405. At step S405, TCB2 set as the link information 94 of TCB1 is copied to the processing-object data-address storage region 105 next time. Thereby, a processing object serves as TCB2 next time. After setting up a processing object next time, processing of a task 2 is performed in step S406. If the wait-condition discharge information 95 on the task which is going to carry out also here and it is going to perform the value of an event flag and after this is in agreement, the event formation waiting state of the task concerned will be canceled. However, the value of the wait-condition discharge information 95 on a task 2 is "2", and since the value of an event flag is

"1", it becomes inharmonious, and TCB1 remains to link structure data, without being canceled of link structure data. After skipping activation of a task 2, it shifts to step S407. Interruption processing is permitted at step S407. Thus, interruption processing should occur immediately after [at the time of permitting interruption processing].

[0096] (Condition (d-2)) At the time of generating of this interruption processing, it shifts to processing of step S801. In step S802, TCB2 is set as the object for a restart here, and the priority and link place (TCB3) for [this] a restart are saved. In step S803, interruption processing is performed after setting out for a restart, and preservation of the link place for a restart. System call wait_flg (task1) should be called at the time of this interruption processing activation. TCB1 is added to link structure data by activation of this wait_flg (task1), and shifts to step S804 by it. Since TCB1 was added, it means that modification had arisen to link structure data, and step S804 serves as Yes. If step S804 serves as Yes, it will shift to step S805. Here, since TCB was inserted, step S805 serves as No and shifts to step S807. Step S807 serves as Yes and shifts to step S808. Step S808 compares a priority with the newly inserted task with the object for a restart. Here, it turns out that TCB in which the direction of TCB1 inserted newly was newly inserted since the priority was high has been arranged before for a restart. Since TCB with a higher priority was inserted, as shown in an arrow head a2 next time by making into a processing object TCB1 which is an object for insertion, at step S809, it sets up, in order to redo processing from TCB newly inserted in link structure data.

[0097] (Condition d-3) Interruption processing is completed by the above processing. Processing of the interrupted side is resumed by termination of interruption processing. Since it is step S407, that interruption processing occurred resumes from step S408. Since TCB1 was set as the processing object here next time by step S408 in interruption processing – step S409, processing of link structure data is redone from TCB1. That is, step S404 – step S409 are repeatedly performed until step S404 is set to Yes.

[0098] In the above-mentioned processing, if TCB is inserted before for a restart, the newly inserted TCB will be set as a processing object next time. Moreover, if TCB is inserted after for a restart, it is not necessary to change a processing object next time. Since processing of the interrupted side is resumed as mentioned above from the next TCB of TCB currently performed at the end in processing of the interrupted side according to this operation gestalt, progress of the processing in processing of the interrupted side can be raised.

[0099] (The 3rd operation gestalt) Although TCB was combined in link structure data with the 2nd operation gestalt, the memory management block is combined with link structure data with the 3rd operation gestalt. The memory management block combined with link structure data here shows free memory block which is not assigned to a task. That is, size is arranged in link structure data at small order, and free memory block waits to be assigned to a task as what shows a free block.

[0100] It searches for the minimum thing from link structure data among free memory block of size which will fill with this condition the demand shown in this get_blk if which task publishes get_blk, and assigns an issuance attaching task. Drawing 13 (a) and (b) are drawings showing an example of quota processing of the memory management block in the 3rd operation gestalt. Drawing 13 (a) and (b) are drawings showing an example of quota processing of the memory management block in the 3rd operation gestalt.

[0101] As shown in drawing 13 (a), when get_blk which asks for free memory block of size "350" is published from which task, retrieval is performed in order of the memory management block BLK1 -> memory management block BLK2 -> memory management block BLK3 -> memory management block BLK4, the memory management block BLK4 assigns, and it is applicable. Thus, interruption processing should occur in the phase which making the memory management block BLK4 applicable to allocation decided. rel_blk (350) should be published in this interruption processing. This rel_blk (350) is a system call of a purport which opens free memory block of size "350", and shows link structure data to drawing 13 (b) by having performed this system call. When drawing 13 (b) is referred to, it turns out that the memory management block BLK5 is inserted between the memory management block BLK3 and the memory management block BLK4. Thus, if free memory block is inserted, the priority of free memory block inserted newly and free memory block saved as an object for a restart will be measured like the 2nd operation gestalt, and it will be determined whether the object for a restart should be updated. The size of the memory management block

BLK4 is "400" here, and the size of the memory management block BLK5 is "350." If it compares by considering such sizes to be priorities, it turns out that the priority of the memory management block BLK5 is high. Therefore, in step S809, the memory management block BLK5 is set as the object for a restart, interruption processing is ended and processing of the interrupted side is resumed.

[0102] Although the processing object was set up in the procedure of the 2nd operation gestalt next time, free memory block set as the link head data-address storage region 92 of the link header 90 may be set as the processing-object data-address storage region 105 like the 1st operation gestalt next time. It may record on the record medium which can computer read the operating system explained in **** 1 – the 3rd operation gestalt, and you may make it the object of a negotiation and a sale. Although there are an IC card, an optical disk, a floppy disk, etc. in such a record medium, utilization is presented with the machine program recorded on these by being installed in the device which has a processor and memory. This device performs the installed machine program serially, and realizes the function of the control-block processor shown in this operation gestalt.

[0103]

[Effect of the Invention] The link structure data with which this invention consists of two or more control block combined using the link information as explained above, The address storage region where the address of which control block is written in among two or more control block, The actuation which decides control block by which the address is written in the address storage region as a processing object this time, Control block of the link place of the control block is decided as a processing object next time. The processing-object decision means which repeats the actuation which writes the address of a processing object in an address storage region the next time concerned about all control block in link structure data, A processing means to perform processing about a processing object this time [this] whenever a processing object is decided this time, It has an authorization means to permit generating of interruption processing in a period after processing about a processing object is performed this time until the following this time processing object is decided. Said processing-object decision means While it is started when interruption processing occurs in the period when interruption processing was permitted, and deciding which control block in link structure data as a processing object this time in interruption processing It decides as a processing object the next time which should be processed in case processing of the side interrupted in another control block is resumed. [when resuming processing of the side which it has the reentrant routine which writes the address of a processing object in an address storage region the next time concerned, and was interrupted after interruption processing generating, and when interruption processing does not occur] It is constituted so that control block currently written in the address storage region may be decided as a processing object this time.

[0104] When resuming processing of the side interrupted after interruption processing generating, in both sides when interruption processing does not occur, a processing object can be decided in a common procedure next time. The definite procedure of a processing object is communalized next time, and since it is simplified, it becomes unnecessary to supervise a flag by processing of the interrupted side like the technique indicated by JP.9-311794.A. Therefore, the time amount which decision of a processing object takes processing-object-next time this time can be stopped to the minimum. If the time amount which processing-object decision takes next time is shortened, the system operation time which can be assigned to the task corresponding to control block will become long. If the system operation time assigned to the task becomes long, since a system designer's burden is mitigated, only the part can design an application program efficiently.

[0105] Moreover, in the above-mentioned equipment, which thing is canceled of link structure data among said two or more control block at the time of interruption processing activation, and said reentrant routine may decide control block which remained in link structure data, without being canceled as a processing object next time, and it may constitute it so that the address may be written in an address storage region.

[0106] Since a processing object is normally set up next time in processing of the side which interrupted when which control block is canceled of link structure data at the time of interruption processing activation, it is not necessary to supervise a flag in processing of the interrupted side. Since the monitor burden of the flag in processing of the interrupted side mitigates, the time

amount to a task switch can make it able to mitigate by the increment in the processing time by the unnecessary flag monitor, and the time amount by which the task equivalent to application is not controlled as a priority can be decreased.

[0107] In the above-mentioned equipment, when interruption processing is materialized [in / in any they are / a control-block processor] among two or more events, it is generated. Moreover, to each of control block The task corresponds and it has the event information which shows the event to which the task corresponding to this is waiting for formation to control block. What was in agreement with the event it was notified in interruption processing among said two or more control block that the event information was After performing at the time of interruption processing activation, it is canceled of link structure data. Said reentrant routine Control block which remained in link structure data, without being canceled may be decided as a processing object next time, and you may constitute so that the address may be written in an address storage region. Thus, if constituted, after an event is materialized, it can control immediately to the task which was waiting for formation of the event, and real-time nature can be raised. Thereby, real-time system can be constituted simply.

[0108] Moreover, in the above-mentioned equipment, said reentrant routine may decide control block used as the link place of control block canceled in link structure data as a processing object next time, and it may constitute it so that the address may be written in an address storage region. Since it becomes unnecessary to process again from the head of link structure data in the interrupted system call processing like the technique indicated by JP,9-311794,A, the increment in the system call processing time by interruption can be stopped to the minimum.

[0109] In the above-mentioned equipment, control block corresponding to which task is inserted in link structure data in the time of interruption processing activation. Moreover, said reentrant routine It has the decision section which determines any of the task corresponding to control block and inserted control block for [which should be performed at the time of the restart to the interrupted side] a restart it should perform first they are. Control block determined by the decision section may be decided as a processing object next time, and you may constitute so that the address may be written in an address storage region.

[0110] In case processing of the interrupted side is resumed, since it is determined flexibly whether the task control block which is the processing side of the side which interrupted and was inserted in link structure data should be processed first, or the task control block which should have been performed next before processing generating of the interrupted side should be processed first, it can raise the processing effectiveness in real-time system.

[0111] Moreover, in the above-mentioned equipment, the priority based on the superiority or inferiority of the task corresponding to it is given to control block, and said decision section measures a priority by control block for [which should be performed at the time of the restart to the interrupted side] a restart, and control block for [which was inserted] insertion, and it may constitute it so that it may determine any of the task corresponding to any should be performed first. Since the priority which faced determining whether the task control block which is the processing side of the side which interrupted and was inserted in link structure data should be first processed in case processing of the interrupted side is resumed, or the task control block which should have been performed next before processing generating of the interrupted side should have been processed first, and was defined beforehand is referred to, a system can operate as a priority.

[0112] Moreover, in the above-mentioned equipment, link structure data have the link header which shows control block corresponding to the task which should be performed to the beginning among two or more tasks, and a reentrant routine may decide control block shown in a link header among control block which remained in link structure data, without being canceled as a processing object next time, and it may be constituted so that the address may be written in an address storage region. Since selection processing of the data set as the memory in which said external reference required in order to maintain the consistency of link structure data in the system call processing called from interruption processing is possible becomes unnecessary, improvement in the speed of interruption processing can be attained.

[0113] Moreover, in the above-mentioned equipment, a control-block processor has two or more tasks and free memory block which is empty memory block, each control block corresponds with

each of two or more free memory block, and said processing means may be constituted so that it may have the allocation section which assigns free memory block corresponding to control block which serves as a processing object this time to which task. Thus, when constituted, free memory block can be efficiently assigned to which task.

[0114] Moreover, in the above-mentioned equipment, which thing is canceled of link structure data among said two or more control block at the time of interruption processing activation, and said reentrant routine may decide control block which remained in link structure data, without being canceled as a processing object next time, and it may constitute it so that the address may be written in an address storage region. Since a processing object is normally set up next time in processing of the side which interrupted when which memory control block is canceled of link structure data at the time of interruption processing activation, processing of the interrupted side does not need to supervise flag information. Since the monitor burden of the flag information in processing of the interrupted side mitigates and the time amount to a task switch mitigates by the increment in the processing time by the unnecessary flag monitor, the time amount by which the task equivalent to application is not controlled as a priority can be decreased.

[0115] Moreover, in the above-mentioned equipment, said reentrant routine may decide control block used as the link place of control block canceled in link structure data as a processing object next time, and it may constitute it so that the address may be written in an address storage region. Since it becomes unnecessary to process again from the head of link structure data in the interrupted system call processing like the technique indicated by JP,9-311794A, the increment in the system call processing time by interruption can be stopped to the minimum.

[0116] Moreover, in the above-mentioned equipment, control block corresponding to which free memory block is inserted in link structure data in the time of said interruption processing activation. It has the decision section which determines any of free memory block corresponding to control block and inserted control block for [which should be assigned at the time of the restart of processing of the interrupted side] a restart said reentrant routine should assign first a task. Control block determined by the decision section may be decided as a processing object next time, and you may constitute so that the address may be written in an address storage region. In case processing of the interrupted side is resumed, since it is determined flexibly whether memory control block which is the processing side of the side which interrupted and was inserted in link structure data should be first assigned to a task, or memory control block which should have been performed next before processing generating of the interrupted side should be first assigned to a task, it can raise the processing effectiveness in real-time system.

[0117] Moreover, in the above-mentioned equipment, the priority based on the memory size of free memory block corresponding to it is given to control block, and said decision section measures a priority by control block for [which should be assigned to a task at the time of the restart to the interrupted side] a restart, and control block for [which was inserted] insertion, and it may constitute it so that it may determine any of free memory block corresponding to any should be first assigned to a task. Since it faces determining whether memory control block which is the processing side of the side which interrupted and was inserted in link structure data when resuming processing of the interrupted side is previously assigned to a task, or memory control block which should have assigned next the task before processing generating of the interrupted side should be processed first and the size of free memory block is referred to, free memory block can assign each task in order of size.

[0118] Moreover, in the above-mentioned equipment, link structure data have the link header which shows the address of control block corresponding to free memory block which should be assigned to the beginning among two or more free memory block at a task, and the 1st allocation section may be constituted so that the address of control block shown in a link header among control block which remained in link structure data, without being canceled may be written in a global field. Since selection processing of the data set as the memory in which said external reference required in order to maintain the consistency of link structure data in the system call processing called from interruption processing is possible becomes unnecessary, improvement in the speed of interruption processing can be attained.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram having shown an example of a computer apparatus which carried the operating system.

[Drawing 2] They are the link header 90 arranged to the external reference field of a memory apparatus 104, and drawing showing an example of link structure data.

[Drawing 3] It is the block diagram showing functionally the configuration of the system call processing section 201 in processing of the side interrupted by interruption processing.

[Drawing 4] It is the flow chart which shows the content of processing of the system call processing section 201 in processing of the side interrupted by interruption processing.

[Drawing 5] It is the block diagram showing the function of the system call processing section 201 which performs interruption processing.

[Drawing 6] It is drawing which added the flow chart of other system call processings in which it interrupted during system call processing to the flow chart of the processing of the interrupted side shown in [drawing 4](#).

[Drawing 7] It is the block diagram showing functionally the configuration of the system call processing section 201 in processing of the side which interrupted.

[Drawing 8] It is drawing which added the flow chart of other system call processings in which it interrupted during system call processing to the flow chart of near processing in which the 1st operation gestalt shown in [drawing 4](#) interrupted.

[Drawing 9] It is drawing showing a signs [that the processing-object data-address storage region 105 is updated next time] condition (c-1), a condition (c-2), a condition (c-3), and a condition (c-4).

[Drawing 10] TCB2-TCB3-TCB4 is arranged at link structure data, and it is drawing showing the condition that TCB1 is set to the link header 90.

[Drawing 11] It is drawing showing signs that link structure data are changed by processing shown in the flow chart of [drawing 8](#).

[Drawing 12] It is drawing showing signs that link structure data are changed by processing shown in the flow chart of [drawing 8](#).

[Drawing 13] It is drawing showing an example of quota processing of free memory block in the 3rd operation gestalt.

[Drawing 14] It is a block diagram for explaining the basic function of OS, and is drawing showing functional block related to execution sequence management of a task, and allocation sequence management of free memory block directly.

[Drawing 15] It is drawing for explaining a reentrant routine.

[Drawing 16] It is drawing showing an example of the link structure data which serve as the link header 90 from two or more Task Control Block (TCB).

[Drawing 17] It is drawing showing signs (a condition (a-1), a condition (a-2), a condition (a-3), condition (a-4) condition (a-7)) that a processing-object storage region is updated this time a processing-object storage region and next time based on the link structure data shown in [drawing 11](#).

[Drawing 18] Two or more tasks shown in link structure data are drawings showing signs that sequential execution is carried out.

[Drawing 19] It is drawing in which the authorization section of interruption processing generating

and the prohibition section were established.

[Drawing 20] It is drawing showing assignment of the memory area at the time of interruption processing generating.

[Drawing 21] the system call of a purport which interruption processing occurs in the interruption processing authorization section, and sets an event flag as "2" in this interruption processing — when "set_flg (2)" is called, they are a processing-object storage region and drawing showing how a processing-object storage region changes next time this time

[Drawing 22] When interruption processing occurs in the interruption processing authorization section and system call "set_flg(2)" is called in this interruption processing, it is drawing showing how link structure data change.

[Drawing 23] Interruption processing occurs in the interruption processing authorization section, and it sets to this interruption processing, and is system call service. When set_flg (2) is called, they are a processing-object storage region and drawing showing how a processing-object storage region changes next time this time.

[Drawing 24] Interruption processing occurs in the interruption processing authorization section, and it sets to this interruption processing, and is system call service. When set_flg (2) is called, link structure data and flag information are drawings showing how it changes.

[Description of Notations]

90 Link Header

92 Link Head Data-Address Storage Region

93 Event Flag Value

94 Link Information

95 Condition Discharge Information

101 Program Control Equipment

102 Input Unit

103 Output Unit

104 Memory Apparatus

105 Next Processing-Object Data-Address Storage Region

111 Input/output Control

112 Processing Section

113 Task

141 System Call Service

201 System Call Processing Section

202 Scheduler Support Task

203 Dispatcher

301 Maine Control Section

302 Interrupt Control Section

303 This Time Processing-Object Data-Address Acquisition Section

304 Data-Processing Termination Judging Section

305 Next Processing-Object Data Decision Section

306 Data-Processing Section

310 This Time Processing-Object Data-Address Storage Region

501 Maine Control Section

502 Data-Processing Section

503 Link Structure Data Modification Judging Section

504 Next Processing-Object Data-Address Initialization Section

701 Maine Control Section

702 Resumption Data Position-Memory Section of Processing

703 Data-Processing Section

704 Link Structure Data Modification Judging Section

705 Resumption Data Fixing Section of Processing

710 Resumption Data Location Storage Region of Processing

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-315162

(P2000-315162A)

(43) 公開日 平成12年11月14日 (2000.11.14)

(51) Int.Cl. ⁷	識別記号	F I	キーワード [*] (参考)
G 0 6 F	9/46	G 0 6 F	3 4 0 B
	3 4 0		5 B 0 9 8
			3 4 0 C

審査請求 未請求 請求項の数26 O L (全 27 頁)

(21) 出願番号 特願平11-123675

(22) 出願日 平成11年4月30日 (1999.4.30)

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 枯木 正吉

東広島市鏡山3丁目10番18号 株式会社松

下電器情報システム広島研究所内

(74) 代理人 100090446

弁理士 中島 司朗 (外1名)

Fターム(参考) 5B098 B306 B308 G404 G302 G314

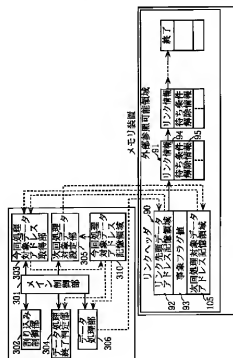
G315 G038 G002 G014

(54) 【発明の名称】 制御ブロック処理装置及び制御ブロック処理プログラムを記録したコンピュータ読み取り可能な記録媒体

(57) 【要約】

【課題】 排他的なリンク構造を持つデータの処理において、割り込み禁止時間を短縮化するとともに、割り込み禁止時間の短縮化を施した場合のシステムコールにかかる処理時間の増加を短縮化する。

【解決手段】 次回処理対象となるデータのアドレスを求めて記憶するための次回処理対象データアドレス記憶領域105を外部参照可能なメモリ領域に設ける。システムコール処理中の一時的に割り込みを許可した時に発生した割り込みにより、割り込まれたシステムコール処理内で扱っていたリンク構造データのリンク構造が他のシステムコール処理により変更されたとき、リンク構造を変更した側のシステムコール処理部において前記次回処理対象データアドレス記憶領域105に変更後のリンク構造データの最適なデータアドレスを設定する。



【特許請求の範囲】

【請求項1】 リンク情報を用いて結合されている複数の制御ブロックからなるリンク構造データと、複数の制御ブロックのうち何れかの制御ブロックのアドレスが書き込まれるアドレス記憶領域と、アドレス記憶領域にアドレスが書き込まれている制御ブロックを今回処理対象として決定する動作と、その制御ブロックのリンク先の制御ブロックを次回処理対象として決定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込む動作とをリンク構造データにおける全ての制御ブロックについて繰り返す処理対象決定手段と、

今回処理対象が決定される度に、この今回処理対象についての処理を行う処理手段と、

今回処理対象についての処理が行われてから、今回処理対象が次に決定されるまでの期間において割り込み処理の発生を許可する許可手段とを備え、

前記処理対象決定手段は、

割り込み処理を許可した期間において割り込み処理が発生した場合に起動され、リンク構造データにおける何れかの制御ブロックを割り込み処理における今回処理対象として決定すると共に、別の制御ブロックを割り込まれた側の処理を再開する際に処理すべき次回処理対象として決定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込むリエントラントルーチンを有し、割り込み処理発生後割り込まれた側の処理を再開する場合、及び、割り込み処理が発生しなかった場合において、アドレス記憶領域にアドレスが書き込まれている制御ブロックを今回処理対象として決定することを特徴とする制御ブロック処理装置。

【請求項2】 前記複数の制御ブロックのうち、何れかのものは、割り込み処理実行時において、リンク構造データから解除され、

前記リエントラントルーチンは、

解除されずにリンク構造データに残った制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項1記載の制御ブロック処理装置。

【請求項3】 割り込み処理は複数の事象のうち何れかが制御ブロック処理装置において成立した際に発生し、制御ブロックのそれぞれには、タスクが対応しており、制御ブロックには、これに対応するタスクが成立を待っている事象を示す事象情報を有しており、前記複数の制御ブロックのうち、その事象情報が割り込み処理において通知された事象と一致したものは、割り込み処理実行時において、実行された後にリンク構造データから解除され、

前記リエントラントルーチンは、

解除されずにリンク構造データに残った制御ブロックを次回処理対象として決定して、そのアドレスをアドレス

記憶領域に書き込むことを特徴とする請求項2記載の制御ブロック処理装置。

【請求項4】 前記リエントラントルーチンは、リンク構造データにおいて解除された制御ブロックのリンク先となる制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項3記載の制御ブロック処理装置。

【請求項5】 割り込み処理実行時において、リンク構造データには何れかのタスクに対応する制御ブロックが挿入され、

前記リエントラントルーチンは、

割り込まれた側への再開時に実行すべき再開対象の制御ブロック及び挿入された制御ブロックに対応するタスクの何れかを先に実行すべきかを決定する決定部を備え、決定部により決定された制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項4記載の制御ブロック処理装置。

【請求項6】 制御ブロックには、それに対応するタスクの優先に基づいた優先度が付与されており、

前記決定部は、

割り込まれた側への再開時に実行すべき再開対象の制御ブロックと、挿入された挿入対象の制御ブロックとで優先度を比較して、何れに対応するタスクの何れを先に実行すべきかを決定することを特徴とする請求項1～5の何れかに記載の制御ブロック処理装置。

【請求項7】 リンク構造データは、複数のタスクのうち、最初に実行すべきタスクに対応する制御ブロックを示すリンクヘッダを有しており、

リエントラントルーチンは、

解除されずにリンク構造データに残った制御ブロックのうち、リンクヘッダに示される制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項6記載の制御ブロック処理装置。

【請求項8】 制御ブロック処理装置は、複数のタスクと、空きのメモリブロックであるフリーメモリブロックとを有し、

各制御ブロックは、複数のフリーメモリブロックのそれぞれと対応しており、

前記処理手段は、

今回処理対象となる制御ブロックに対応するフリーメモリブロックを、何れかのタスクに割り当てる割当部を備えることを特徴とする請求項1記載の制御ブロック処理装置。

【請求項9】 前記複数の制御ブロックのうち、何れかのものは、割り込み処理実行時において、リンク構造データから解除され、

前記リエントラントルーチンは、

解除されずにリンク構造データに残った制御ブロックを

次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項8記載の制御ブロック処理装置。

【請求項10】 前記リエントラントルーチンは、リンク構造データにおいて解除された制御ブロックのリンク先となる制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項9記載の制御ブロック処理装置。

【請求項11】 前記割り込み処理実行時において、リンク構造データには何れかのフリーメモリブロックに対応する制御ブロックが挿入され、前記リエントラントルーチンは、割り込まれた側の処理の再開時に割り当てるべき再開対象の制御ブロック及び挿入された制御ブロックに対応するフリーメモリブロックの何れを先にタスクに割り当てるべきかを決定する決定部を備え、

決定部により決定された制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項8～10の何れかに記載の制御ブロック処理装置。

【請求項12】 制御ブロックには、それに対応するフリーメモリブロックのメモリサイズに基づいた優先度が付与されており、

前記決定部は、割り込まれた側への再開時にタスクに割り当てるべき再開対象の制御ブロックと、挿入された挿入対象の制御ブロックとで優先度を比較して、何れに対応するフリーメモリブロックの何れを先にタスクに割り当てるべきかを決定することを特徴とする請求項11記載の制御ブロック処理装置。

【請求項13】 リンク構造データは、複数のフリーメモリブロックのうち、最初にタスクに割り当てるべきフリーメモリブロックに対応する制御ブロックのアドレスを示すリンクヘッダを有しており、

第1割当部は、解除されずにリンク構造データに残った制御ブロックのうち、リンクヘッダに示される制御ブロックのアドレスを大域的領域に書き込むことを特徴とする請求項12記載の制御ブロック処理装置。

【請求項14】 制御ブロック処理プログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記コンピュータは、

リンク情報を用いて結合されている複数の制御ブロックからなるリンク構造データと、複数の制御ブロックのうち何れかの制御ブロックのアドレスが書き込まれるアドレス記憶領域とを備え、前記制御ブロック処理プログラムは、アドレス記憶領域にアドレスが書き込まれている制御ブロックを今回処理対象として決定する動作と、当該制御ブロックのリンク先の制御ブロックを次回処理対象とし

て決定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込む動作とをリンク構造データにおける全ての制御ブロックについて繰り返す処理対象決定ステップと、

今回処理対象が決定される度に、この今回処理対象についての処理を行う処理ステップと、今回処理対象についての処理が行われてから、今回処理対象が次に決定されるまでの期間において割り込み処理の発生を許可する許可ステップとを備え、

前記処理対象決定ステップは、割り込み処理を許可した期間において割り込み処理が発生した場合に起動され、リンク構造データにおける何れかの制御ブロックを割り込み処理における今回処理対象として決定すると共に、別の制御ブロックを割り込まれた側の処理を再開する際に処理すべき次回処理対象として決定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込むリエントラントルーチンを有し、割り込み処理発生後割り込まれた側の処理を再開する場合、及び、割り込み処理が発生しなかった場合において、アドレス記憶領域にアドレスが書き込まれている制御ブロックを今回処理対象として決定することを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項15】 前記複数の制御ブロックのうち、何れかのものは、割り込み処理実行時において、リンク構造データから解除され、

前記リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項14記載のコンピュータ読み取り可能な記録媒体。

【請求項16】 割り込み処理は複数の事象のうち何れかが制御ブロック処理記録媒体のプログラムにおいて成立した際に発生し、制御ブロックのそれぞれには、タスクが対応しており、制御ブロックには、これに対応するタスクが成立を待っている事象を示す事象情報を有しており、

前記複数の制御ブロックのうち、その事象情報が割り込み処理において通知された事象と一致したものは、割り込み処理実行時において、実行された後にリンク構造データから解除され、

前記リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項15記載のコンピュータ読み取り可能な記録媒体。

【請求項17】 前記リエントラントルーチンは、リンク構造データにおいて解除された制御ブロックのリンク先となる制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項16記載のコンピュータ読み取り可能

な記録媒体。

【請求項18】 割り込み処理実行時において、リンク構造データには何れかのタスクに対応する制御ブロックが挿入され、

前記リエントラントルーチンは、

割り込まれた側への再開時に実行すべき再開対象の制御ブロック及び挿入された制御ブロックに対応するタスクの何れかを先に実行すべきかを決定する決定サブステップを備え、

決定サブステップにより決定された制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項17記載のコンピュータ読み取り可能な記録媒体。

【請求項19】 制御ブロックには、それに対応するタスクの優先度に基づいた優先度が付与されており、前記決定サブステップは、

割り込まれた側への再開時に実行すべき再開対象の制御ブロックと、挿入された挿入対象の制御ブロックとで優先度を比較して、何れに対応するタスクの何れかを先に実行すべきかを決定することを特徴とする請求項14～18の何れかに記載のコンピュータ読み取り可能な記録媒体。

【請求項20】 リンク構造データは、複数のタスクのうち、最初に実行すべきタスクに対応する制御ブロックを示すリンクヘッダを有しており、

リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックのうち、リンクヘッダに示される制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項19記載のコンピュータ読み取り可能な記録媒体。

【請求項21】 コンピュータ読み取り可能な記録媒体は、複数のタスクと、空きのメモリブロックであるフリーメモリブロックとを有し、

各制御ブロックは、複数のフリーメモリブロックのそれぞれと対応しており、前記処理ステップは、

今回処理対象となる制御ブロックに対応するフリーメモリブロックを、何れかのタスクに割り当てる割当サブステップからなることを特徴とする請求項14記載のコンピュータ読み取り可能な記録媒体。

【請求項22】 前記複数の制御ブロックのうち、何れかのものは、割り込み処理実行時において、リンク構造データから解除され、

前記リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項21記載のコンピュータ読み取り可能な記録媒体。

【請求項23】 前記リエントラントルーチンは、

リンク構造データにおいて解除された制御ブロックのリンク先となる制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項22記載のコンピュータ読み取り可能な記録媒体。

【請求項24】 前記割り込み処理実行時において、リンク構造データには何れかのフリーメモリブロックに対応する制御ブロックが挿入され、

前記リエントラントルーチンは、

割り込まれた側の処理の再開時に割り当てるべき再開対象の制御ブロック及び挿入された制御ブロックに対応するフリーメモリブロックの何れかを先にタスクに割り当てるべきかを決定する決定サブステップを備え、

決定サブステップにより決定された制御ブロックを次回処理対象として決定して、そのアドレスをアドレス記憶領域に書き込むことを特徴とする請求項21～23の何れかに記載のコンピュータ読み取り可能な記録媒体。

【請求項25】 制御ブロックには、それに対応するフリーメモリブロックのメモリサイズに基づいた優先度が付与されており、

前記決定サブステップは、

割り込まれた側への再開時にタスクに割り当てるべき再開対象の制御ブロックと、挿入された挿入対象の制御ブロックとで優先度を比較して、何れに対応するフリーメモリブロックの何れかを先にタスクに割り当てるべきかを決定することを特徴とする請求項24記載のコンピュータ読み取り可能な記録媒体。

【請求項26】 リンク構造データは、複数のフリーメモリブロックのうち、最初にタスクに割り当てるべきフリーメモリブロックに対応する制御ブロックのアドレスを示すリンクヘッダを有しており、

第1割当サブステップは、

解除されずにリンク構造データに残った制御ブロックのうち、リンクヘッダに示される制御ブロックのアドレスを大域的領域に書き込むことを特徴とする請求項25記載のコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、リンク情報を用いて結合されている複数の制御ブロックに対して処理を行う制御ブロック処理装置及び制御ブロック処理プログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】高度情報化社会の余波は家電業界にまで押し寄せており、将来における家電製品の高機能化の要望に応じるため、これら家電製品向けの基盤ソフトウェアの整備を行うことや、即時応答性が高いオペレーティングシステムの開発を行うことが急務になりつつある（オペレーティングシステムをOSと略記する。また即時

応答性が求められるOSをリアルタイムシステムという)。

【0003】このようなリアルタイムシステムには、汎用コンピュータ向けのOSと同様、複数タスクの実行順序管理やフリーメモリブロックの割当順序管理といったシステムの基盤制御が求められる。ここで複数タスク実行順序管理とは、メモリに複数のタスクがロードされている場合、各タスクの優先度の高低や実行待ち状態になった時刻順に基づいて各タスクをどのような順序で実行してゆくかを定めて、その実行順序に応じて順次起動してゆくという基盤制御である。

【0004】複数フリーメモリブロックの割当順序管理とは、メモリ上の空き領域に相当するフリーメモリブロックをどの順序でタスクに割り当てるかを定めて、タスクが要求するメモリサイズに応じて、複数のフリーメモリブロックのうち何れかを、要求元のタスクに割り当ててゆくという基盤制御である。また家電機器等における状態変化は、割り込み処理により通知されるので、リアルタイムシステムでは、割り込み処理に対しての即時応答性を保証しつつも複数タスクの実行順序管理やフリーメモリブロックの割当順序管理を行わねばならない。

【0005】以下、従来のOSについて説明する。図14は、OSの基本機能を説明するためのブロック図であり、タスクの実行順序管理、フリーメモリブロックの割当順序管理に直接関係する機能ブロックを示したもので、他の部分は省略してある。割り込み処理部112は、図示しない入力機器などからの割り込みを受け、それにに応じて必要であるならばOSの提供するシステムコールサービス141を呼び出す。

【0006】システムコールサービスとは、OSから提供される同期制御、排他制御、タスク間通信機能、メモリ管理機能などの基本的なルーチン処理である。呼び出されたシステムコールサービス141はシステムコール処理部201にて対応する処理を実行し、このとき必要であればタスクのステータスを変更する。システムコール処理部201にてタスクステータスが変更された場合、割り込み処理によりタスクのステータス変更が発生したという事象をタスクスケジューラ202に通知する。タスクステータスとは、タスクが入出力を待っている状態などを示す状態フラグであり、このステータスが変化すること、事象待ち状態のタスクを実行可能状態に移したり、実行されていたタスクが事象待ち状態に移することを意味する。尚、タスク管理に関するシステムコール、フリーメモリブロックに関するシステムコールには、`[rel_wait]`、`[set_flg]`、`[wait_flg]`、`[rel_blk]`といったものがある。

【0007】システムコール`set_flg()`とは、事象フラグがある値に設定されたことを通知するシステムコールである。システムコール`rel_wait()`とは、あるタスクの事象成立待ち状態を強制解除するシステムコールであ

る。システムコール`wait_flg()`とは、任意のタスクを事象成立待ち状態として新たにエントリーするシステムコールである。

【0008】システムコール`rel_blk`は、任意のサイズのフリーメモリブロックを開放する旨のシステムコールである。タスクスケジューラ202は、割り込みによるタスクの状態変化を反映し、タスク全体の実行順序新たに決定し、ディスパッチャ203にタスクの切り換えを通知する。

【0009】ディスパッチャ203は現在の実行タスクを中断し、タスクスケジューラ202により決定したタスクを新たな実行タスクに切り替える。即時応答性が要求されるリアルタイムシステムでは、タスクに緊急度に応じた優先度を付けておき、タスクスケジューラ202がこの優先度を元に、優先度の高いタスクから順に実行するようスケジューリングを行う。

【0010】通常、使用メモリ容量の節約などから、個々の割り込み処理やタスクに対して専用のOSが用意されているわけではなく、OSは割り込み処理およびタスクの共通ルーチンとなっている。これはシステムコールが呼び出され処理が完了する前に、同一のシステムコールが呼び出される可能性があることを意味する。このような性質をもつルーチンをリエントラントルーチン(再入可能性を有するルーチン)と呼ぶ。

【0011】図15は、リエントラントルーチンを説明するための図である。図15(a)はリエントラントではないシステムコールの例であり、図15(b)(c)はリエントラントルーチンの例である。図15(a)は、リエントラントにしないため、システムコール処理を全割り込み禁止としていることにより、割り込み発生と割り込み処理開始との間に時間のずれが生じている。図15(b)は、タスクAのシステムコール処理中に外部割り込みが発生し、割り込み処理から同一システムコールの呼び出しが行われる例を示している。図15(c)は、タスクAのシステムコールが処理されている間に外部割り込みが発生し、割り込み処理部から呼び出された他のシステムコールによりタスクBの切り換えが発生し、その切り換わったタスクBからの割り込みにより中断されたシステムコールの呼び出しが行われる例である。

【0012】複数タスクの実行順序管理やフリーメモリブロックの割当順序管理は、このようなリエントラントという性質を持つシステムコールによって行われるのが理想的である。また、複数タスクの実行順序管理やフリーメモリブロックの割当順序管理はリンク構造データを用いることにより管理されている点で実行される。従って複数タスクの実行順序管理に用いられるリンク構造データの一例について説明する。

【0013】図16は、リンクヘッダ90と、複数のTask Control Block(TCB)からなるリンク構造データの一例

例を示す図である。リンクヘッダ90はイベントフラグ管理ブロックとも呼ばれ、複数のTCBのうち、最初に実行すべきTCBのアドレスが設定されるリンク先頭データアドレス記憶領域92と、リアルタイムシステムにおいて成立している事象を示す値が設定される事象フラグ値93とを有する。

【0014】3つのTCB91は、実行待ち状態にある3つのタスク(タスク1、タスク2、タスク3)のそれぞれに対応するものであり、自身に対応するタスクを実行した後、次に実行すべきタスクのTCBを示すリンク情報94と、このタスクの実行待ち状態が解除されるための条件を示す待ち条件解除情報95とを有する。図16の一例において、リンクヘッダ90→リンク情報94のリンク関係参照すると、TCB1のリンク情報94はTCB2と設定され、TCB2のリンク情報94はTCB3、TCB3のリンク情報94は“終了”と設定されている。

【0015】図16のリンク構造データは、TCB1→TCB2→TCB3→“終了”といった順に実行順序が与えられていることがわかる。この実行順序は、各タスクについての優先度の高低や実行待ち状態になった時刻順に基づいて定められている。またTCB1における待ち条件解除情報95は“1”と設定され、TCB2における待ち条件解除情報95は“2”と設定されているので、TCB1、TCB2は事象フラグ値93が“1”、“2”という値に設定されるのを待っており、未だ何れのTCBも実行待ち状態にあることがわかる。ここで事象フラグ値93が“1”になれば、TCB1の事象成立待ち状態が解除され、事象フラグ値93が“2”になればTCB2の事象成立待ち状態が解除される。このようなリンク構造データは外部参照領域に配置される。ここで外部参照領域とは、大域の領域とも呼ばれ、システムコール、割り込み処理、タスクの何れもが共通にアクセスすることができるメモリ領域である。外部参照領域に対して、システムコール、割り込み処理、タスクのそれぞれが専用にアクセスすることができるメモリ領域を局所的領域という。

【0016】なお、リンク構造データは、必ずしも前に示したようなリンク情報にて接続されているデータ構造だけではない。例えば可変長のデータが連続して配置されており、各データのサイズ情報により任意データに対する次データが求まるようなデータ構造もリンク構造データである。すなわち、任意の処理対象データが求まることにより、次の処理対象データが求まるようなデータ構造をリンク構造データと呼ぶ。

【0017】続いて、図16に示すリンク構造データを用いて複数タスクを順次実行してゆく際の手順について説明する。ここで留意すべきは、システムコール処理部201が局所的領域に今回処理対象記憶領域、次回処理対象記憶領域を設けて、今回起動すべきタスクと、次に起動すべきタスクとを特定する点である。図17は、図16に示したリンク構造データに基づいて、今回処理

対象記憶領域と、次回処理対象記憶領域とが更新されてゆく様子(状態(a-1)、状態(a-2)、状態(a-3)、状態(a-4)……状態(a-7))を示す図であり、図18は、リンク構造データに示される複数のタスクが順次実行されてゆく様子を示す図である。これらの図において共通の符号(a-1)(a-2)(a-3)……は、共通の状態を意味するものとする。

【0018】(状態(a-1))システムコール処理部201は、リンクヘッダ90におけるリンク先頭データアドレス記憶領域92に示されているTCBのアドレスを今回処理対象記憶領域に設定する。図17の一例では、リンクヘッダ90のリンク先頭データアドレス記憶領域92がTCB1と設定されているので、今回処理対象記憶領域はTCB1と設定される。これにより今回処理対象となるタスクはタスク1と決定されたので、続いてタスク1の次に実行すべきタスクを決定する。ここでタスク1に対応するTCB1のリンク情報94はTCB2と設定されており、今回はこのTCB2に対応するタスク2を処理せねばならない。そこで、次回処理対象記憶領域にTCB2を設定して処理を終える。

【0019】(状態(a-2))今回処理対象となったTCB1の待ち条件解除情報95がどのような事象に設定されているかを参照して、タスク1の事象成立待ち状態を解除するか維持するかを決定する。ここでTCB1の待ち条件解除情報95は“1”と設定されており、リンクヘッダ90の事象フラグ値93は“1”と設定されているので、タスク1が待っていた事象が成立したことがわかる。待っていた事象が成立したので、タスク1を実行すると共にリンク構造データからTCB1を解除する。この解除時点において、TCB1におけるリンク情報94は不定となる。また、TCB1がリンク構造データから解除されたので、リンクヘッダ90におけるリンク先頭データアドレス記憶領域92をTCB2に設定する。

【0020】仮に次回処理対象記憶領域に次回処理対象を設定する前にTCB1をリンク構造データから解除してしまえば、次回処理対象記憶領域の設定値が不定となる。これを避ける意味で、TCB1をリンク構造データから解除する前に次回処理対象記憶領域を設定している。

(状態(a-3))システムコールサービス部201は、状態(a-2)において次回処理対象記憶領域に示されているTCBのアドレスを今回処理対象記憶領域に設定する。図17の状態(a-1)では、次回処理対象記憶領域にTCB2のアドレスが設定されているので、今回処理対象記憶領域はTCB2と設定される。これにより今回処理対象となるタスクはタスク2と決定されたので、続いてタスク2の次に実行すべきタスクを決定する。ここでタスク2に対応するTCB2のリンク情報94はTCB3と設定されており、今回はこのTCB3に対応するタスク3を処理せねばならない。そこで、次回処理対象記憶領域にTCB3を設定して処理を終える。

【0021】(状態(a-4))今回処理対象となったTCB2の待ち条件解除情報95がどのような事象に設定されているかを参照して、タスク2の事象成立待ち状態を解除するか、或は維持するかを決定する。ここでTCB2の待ち条件解除情報95は"2"と設定されており、リンクヘッダ90の事象フラグ値93は"1"と設定されているので、タスク2が待っていた事象は不成立であることがわかる。事象不成立なので、タスク2の実行をスキップする。実行スキップにより、リンクヘッダ90におけるリンク先頭データアドレス記憶領域92の設定はTCB2のま

【0022】(状態(a-5))システムコールサービス部201は、状態(a-3)において次回処理対象記憶領域に示されているTCBのアドレスを今回処理対象記憶領域に設定する。図17の状態(a-3)では、次回処理対象記憶領域にTCB3のアドレスが設定されているので、今回処理対象記憶領域はTCB3と設定される。これにより今回処理対象となるタスクはタスク3と決定されたので、続いてタスク3の次に実行すべきタスクを決定する。ここでタスク3に対応するTCB3のリンク情報94は終了と設定されており、次回で、リンク構造データにおけるタスク実行を終了せねばならない。そこで、次回処理対象記憶領域に終了を設定して処理を終える。

【0023】(状態(a-6))今回処理対象となったTCB3の待ち条件解除情報95がどのような事象に設定されているかを参照して、タスク3の事象成立待ち状態を解除するか、或は維持するかを決定する。ここでTCB3の待ち条件解除情報95は"1"と設定されており、リンクヘッダ90の事象フラグ値93は"1"と設定されているので、タスク3が待っていた事象は成立したことがわかる。条件一致なので、タスク3を実行し、TCB3をリンク構造データから解除する。この解除時点において、TCB3におけるリンク情報94は不定となる。リンクヘッダ90におけるリンク先頭データアドレス記憶領域92はTCB2に設定したまま状態(a-7)に移行して、処理を終了する。

【0024】以上のように各タスクを実行する場合、今回処理対象記憶領域、次回処理対象記憶領域、リンクヘッダ90のリンク先頭データアドレス記憶領域92が不定である状態において、割り込み処理が発生するのは望ましくない。即ち、割り込み処理は今回処理対象記憶領域と、次回処理対象記憶領域と、リンクヘッダ90のリンク先頭データアドレス記憶領域92とが不定となる状態(a-1)、状態(a-2)、状態(a-3)、状態(a-4)、状態(a-5)、状態(a-6)において禁止すべきであり、これらが確定している①②③の区間において許可すべきであることがわかる。

【0025】図19は、以上の観点により、割り込み処理発生時の許可区間、禁止区間を設けた領域である。また図20は、割り込み処理発生時のメモリ領域の割り当てを示す図である。図20において、リンクヘッダ90及び

リンク構造データは外部参照領域に配置されており、割り込まれた側の処理、割り込んだ側の処理がそれぞれ独自にシステムコールサービス呼び出ししており、これがそれぞれ局所的領域を確保している。確保された局所的領域において、今回処理対象記憶領域と、次回処理対象記憶領域とが確保されていることがわかる。

【0026】続いて、割り込み処理に対する即時応答性を保証しながら複数タスクの実行順序管理やフレームモリブロックの割当順序管理を行うには、どのような点に留意せねばならないかを説明する。割り込み処理を許可する場合、複数タスクの実行順序管理にあたってもう一つのことに留意せねばならない。それは、割り込み処理許可区間において割り込み処理が発生した場合、その割り込み処理においてもシステムコールが行われ、このシステムコールにより、リンク構造データが変更される可能性があることである。割り込み処理においてリンク構造データが変更されれば、割り込まれた側の局所的領域における今回処理対象記憶領域と、次回処理対象記憶領域と、割り込み処理において変更されたリンク構造データとが不一致となり、リアルタイムシステムの誤動作を招いてしまう。

【0027】以降、局所的領域内の今回処理対象記憶領域と、次回処理対象記憶領域と、割り込み処理時により変更されたリンク構造データとの不一致がどのようにして生じるかについて説明する。図21は、この割り込み処理許可区間において割り込み処理が発生し、この割り込み処理において事象フラグを"2"に設定する旨のシステムコール"set_flg(2)"が呼び出された場合に今回処理対象記憶領域、次回処理対象記憶領域がどのように変化するかを示す図であり、図22は、割り込み処理許可区間において割り込み処理が発生し、この割り込み処理においてシステムコール"set_flg(2)"が呼び出された場合にリンク構造データがどのように変化するかを示す図である。

【0028】図21、図22において状態(a-1)～状態(a-4)におけるリンク構造データ、今回処理対象記憶領域、次回処理対象記憶領域の内容は図17、図18と同一であるが、図17における①の状態で割り込み処理が発生したため、状態(a-2)の後に状態(b-1)～状態(b-5)が発生し、この状態(b-1)～状態(b-5)の発生により、割り込まれた側の処理の処理が状態(a-5)～状態(a-7)に示すものから、状態(a-8)～状態(a-9)に示すものに置き換わっている。

【0029】以下これらの状態について説明する。(状態(b-1))割り込み処理が発生したので、システムコール処理部201は割り込み処理向けに局所的領域を確保し、ここに今回処理対象記憶領域と、次回処理対象記憶領域とを設ける。この割り込み処理においてset_flg(2)が呼び出されたので、事象フラグ値93は"1"から"2"に更新されている。また図21の状態(a-1)において

リンク先頭データアドレス記憶領域92にTCB2のアドレスが設定されているので、新たに確保された局所的領域における今回処理対象記憶領域はTCB2と設定される。タスク2に対応するTCB2のリンク情報94はTCB3と設定されており、今回はこのTCB3に対応するタスク3を処理せねばならない。そこで、次回処理対象記憶領域にTCB3を設定して処理を終える。

【0030】(状態(b-2))今回処理対象となったTCB2の待ち条件解除情報95がどのような事象に設定されているかを参照して、タスク2の事象成立待ち状態を解除するか、或は維持するかを決定する。ここでTCB2の待ち条件解除情報95は“2”と設定されており、リンクヘッダ90の事象フラグ値93は“2”と設定されているので、タスク2が待っていた事象は成立していることがわかる。条件一致なので、タスク2を実行した後、TCB2をリンク構造データから解除する。その後、リンクヘッダ90におけるリンク先頭データアドレス記憶領域92をTCB3のアドレスに設定する。

【0031】(状態(b-3))システムコールサービス部201は、状態(b-2)において次回処理対象記憶領域に示されているTCBのアドレスを今回処理対象記憶領域に設定する。状態(b-2)では、次回処理対象記憶領域にTCB3のアドレスが設定されているので、今回処理対象記憶領域はTCB3と設定される。これにより今回処理対象となるタスクはタスク3と決定されたので、続いてタスク3の次に実行すべきタスクを決定する。ここでタスク3に対応するTCB3のリンク情報94は終了と設定されており、次回で、リンク構造データにおけるタスク実行を終了せねばならない。そこで、次回処理対象記憶領域に終了を設定して処理を終える。

【0032】(状態(b-4))今回処理対象となったTCB3の待ち条件解除情報95がどのような事象に設定されているかを参照して、タスク3の事象成立待ち状態を解除するか、或は維持するかを決定する。ここでTCB3の待ち条件解除情報95は“1”と設定されており、リンクヘッダ90の事象フラグ値93は“2”と設定されているので、タスク3が待っていた事象は不成立であることがわかる。条件不成立なので、タスク3を実行せず、TCB3をリンク構造データから解除しない。これにより、リンクヘッダ90におけるリンク先頭データアドレス記憶領域92はTCB3に設定されたまま状態(b-5)において処理を終了する。

【0033】(状態(a-8))状態(b-5)において割り込み処理を終了したので、割り込まれた側の処理の状態(a-2)の直後の状態(状態(a-3))に移行する。ここで割り込まれた側の処理におけるシステムコールサービス部201において、今回処理対象記憶領域と、次回処理対象記憶領域とは状態(a-1)において設定された内容がそのまま保存されている。即ち、今回処理対象記憶領域と、次回処理対象記憶領域とは、それぞれTCB1-TCB2に設定

されているのである。

【0034】(状態(a-9))割り込まれた側の処理のシステムコールサービス部201は、今回処理対象記憶領域と、次回処理対象記憶領域との設定内容に従って今回処理対象、次回処理対象を設定しようとする。ここで次回処理対象記憶領域の値を今回処理対象記憶領域にコピーすることにより、今回処理対象記憶領域はTCB2のアドレスに設定される。続いて、次回処理対象記憶領域の値を設定しようとするが、割り込んだ側の処理において、リンク構造データからTCB2は解除されているので、次回処理対象記憶領域には次に実行すべきタスクのTCBを設定することはできない。よって次回処理対象記憶領域における内容は不定値となる。

【0035】既にTCB2はリンク構造データから解除され、タスク2はメモリから除外されているので、TCB2に対応するタスク2を実行しようとする、システムは処理不能に陥ってしまい、システムの誤動作が発生する。以上のようにリンク先頭データアドレス記憶領域92や今回処理対象記憶領域、次回処理対象記憶領域の確定区間であっても、割り込み処理においてリンク構造データが変更されれば、今回処理対象記憶領域と、次回処理対象記憶領域と、割り込み処理時により変更されたリンク構造データとの不一致が発生する恐れがあり、誤動作の恐れがある。

【0036】これは、リンク構造データを扱うシステムコールをリエントラントな処理にする場合、一方の処理においてリンクに関する情報を保持していても、他方の処理によりリンク構造が変更されてしまうため、保持していたリンクに関する情報の整合性が保たなくなることの意味する。この問題は、リエントラントなシステムコール中の割り込み発生により同一のシステムコールを呼び出した場合だけでなく、リエントラントなシステムコール処理中に扱っていたリンク構造データを変更するような他のシステムコールが割り込み発生により呼び出された場合でも同様に起こる。したがって、従来のOSでは、リンク構造データの処理中にリンク構造が変更されないようにするため、リンクに繋がっている全データの処理が完了するまで外部割り込みを禁止するという割り込み制御方式を使用していた。

【0037】しかし、この割り込み制御方式では、リンクに繋がっているデータ数が増えるほど割り込み禁止時間が長くなり、図15(a)に示すように実際の割り込み発生と割り込み処理開始との間に発生する時間のずれが大きくなるという問題がある。リアルタイムシステムは外部割り込みへの即時応答性が重要な意味を持つので、本システムにおいて割り込み禁止時間が良いということは、システムに致命的な欠陥を与える原因となりうる。直接的には、外部割り込みによる通信データの取りこぼし等の問題がある。間接的には、割り込みが遅延されたことによるタスク切り換えの遅延により、優先度通

りにシステムが動作しない時間が増加する等という問題がある。

【0038】したがってOSを使用するアプリケーションは、リンク構造データに繋がる最大数を処理する場合の割り込み禁止時間を考慮してシステム設計を行う必要がある。また、場合によっては、その最大割り込み禁止時間でのシステム構築が不可能となり、専用のハードウェアを備える等の対処方法が必要となる。OSのリンク構造データ処理時における割り込み禁止時間の増加という問題を解決するため、「特開平9-311794」に示す技術では、リンクヘッダに対応したリンク構造データの変更を通知するためのフラグを設けることを提案している。すなわち、この従来技術は、割り込んだ側のシステムコール処理部がリンク構造データを変更した場合、変更を示す値をフラグに設定する。割り込まれた側のシステムコール処理部は、割り込み許可区間においてフラグの変化を監視しており、フラグがシステムコールが変更を示す値に設定されたことを検出すると、リンクの先頭データから処理をやり直すことで、リンクデータが変更されても正しい結果が得られることを保証する。この従来技術を用いれば、割り込み禁止時間を短縮化することが可能となる。

【0039】図23は、割り込み処理許可区間において割り込み処理が発生し、この割り込み処理においてシステムコールサービス set_flg(2)が呼び出された場合に今回処理対象記憶領域、次回処理対象記憶領域がどのように変化するかを示す図であり、図24は、割り込み処理許可区間において割り込み処理が発生し、この割り込み処理においてシステムコールサービス set_flg(2)が呼び出された場合にリンク構造データ、フラグ情報とがどのように変化するかを示す図である。これらの図において共通の符号(a-1)(a-2)(a-3)・・・は共通の状態を意味するものとする。

【0040】図24が図22と異なるのは、図24のリンク構造データにおいてフラグ情報が設けられており、当該フラグ情報が状態(b-2)において「1」に設定されている点である。この状態(b-2)では、TCBがリンク構造データから解除されており、この解除によってリンク構造データのデータ構造が変更したので、この変更を示す意味でフラグ情報が「0」から「1」に変更されたのである。

【0041】図23が図21と異なるのは、割り込んだ側の処理から割り込まれた側の処理にリターンした際以降の処理が状態(a-9)から状態(a-10)、状態(a-11)に置き換えられている点である。

(状態(a-8)) 状態(b-5)において処理を終了したので、割り込まれた側の処理の状態(a-2)の直後の状態(状態(a-8))に移行する。ここで割り込まれた側の処理におけるシステムコールサービス部201において、今回処理対象記憶領域と、次回処理対象記憶領域とは状態(a-1)において設定された内容がそのまま保存されている。

即ち、今回処理対象記憶領域と、次回処理対象記憶領域とは、それぞれTCB1—TCB2に設定されているのである。

【0042】(状態(a-10)) システムコールサービス部201は、リンク構造データにおけるフラグを参照して割り込まれた側の処理においてリンク構造データに変更が生じたかを判定する。ここでリンク構造データのフラグは、リンク構造データのデータ構造に変更があった旨を示す「1」に設定されている。

【0043】(状態(a-11)) リンク構造データのフラグは、変更があった旨を示しているので、割り込まれた側の処理のシステムコールサービス部201は、リンクヘッダ90におけるリンク先頭データアドレス記憶領域92の内容を今回処理対象記憶領域にコピーすることにより、今回処理対象記憶領域をTCB3に設定し、TCB3のリンク情報94に示されているアドレスを次回処理対象記憶領域に設定する。TCB3はリンク構造データに現実存在しており、メモリにもタスク3が存在しているため、このタスク3は正常に実行されることになる。

【0044】
20 【発明が解決しようとする課題】しかしながらこの従来技術では、割り込まれる側のシステムコールにおいて、ひとつのTCBを処理するたびにフラグの監視が必要であり、今回処理対象一回処理対象を確定するために要する時間が「フラグ監視処理に要する時間×データ数」だけ長くなる。図19の一例でいうと、①、②、③、④に相当する許可区間の全てにおいて、フラグ情報が更新されたか否かを監視せねばならないので、このフラグ監視に多大な時間を費やしてしまう。即ち、「特開平9-311794」に示す先行技術においては、割り込みへの即時応答性が向上したものの、割り込みが発生しない場合においては、不要なフラグ監視時間の増加に基づく処理遅延が顕著になってしまふ。

【0045】今回処理対象一回処理対象の確定に要する時間が長くなれば、TCBに対応するタスクに割り当て可能なシステム移動時間が短くなる。タスクに割り当てられたシステム移動時間が短くなれば、その分だけ、アプリケーションプログラムに対しての制約が厳しくなり、リアルタイムシステムの設計が困難になってしまふ。

40 【0046】本発明の目的は、フラグ設定値の監視を必要とせずに、割り込み処理の禁止時間を短縮化することができる制御ブロック処理装置及び制御ブロック処理プログラムを記録したコンピュータ読み取り可能な記録媒体を提供することである。

【0047】
50 【課題を解決するための手段】上記目的を達成するために本発明は、リンク情報を用いて結合されている複数の制御ブロックからなるリンク構造データと、複数の制御ブロックのうち何れかの制御ブロックのアドレスが書き込まれたアドレス記憶領域と、アドレス記憶領域にアド

レスが書き込まれている制御ブロックを今回処理対象として決定する動作と、その制御ブロックのリンク先の制御ブロックを次回処理対象として決定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込む動作とをリンク構造データにおける全ての制御ブロックについて繰り返す処理対象決定手段と、今回処理対象が決定される度に、この今回処理対象についての処理を行う処理手段と、今回処理対象についての処理が行われてから、今回処理対象が次に決定されるまでの期間において割り込み処理の発生を許可する許可手段とを備え、前記処理対象決定手段は、割り込み処理を許可した期間において割り込み処理が発生した場合に起動され、リンク構造データにおける何れかの制御ブロックを割り込み処理における今回処理対象として決定すると共に、別の制御ブロックを割り込まれた側の処理を再開する際に処理すべき次回処理対象として決定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込むリエントラントルーチン有し、割り込み処理発生後割り込まれた側の処理を再開する場合、及び、割り込み処理が発生しなかった場合において、アドレス記憶領域にアドレスが書き込まれている制御ブロックを今回処理対象として決定することを特徴としている。

【0048】

【発明の実施の形態】以下、図面を参照しながら、制御ブロック処理装置の実施形態を説明する。ここで、現状において当業者で慣用されている制御ブロック処理装置の流通・売買の形態は、制御ブロック処理装置と等価な機能を有するオペレーティングシステムの実行形式のソフトウェアを記録媒体に記録して、パッケージソフトとして流通・売買する形態である。このようなパッケージソフトはこれを購買した顧客によりプロセッサやメモリを有する機器にインストールされ、インストールされた機器がこのソフトウェア通りの処理を行うことにより機器は制御ブロック処理装置としての機能を果たす。上記のような形態が慣用されることを考えると、制御ブロック処理装置としての機能主体は、プロセッサ、メモリ等のハードウェア資源というよりは、それらハードウェア資源を用いて制御ブロック処理装置としての処理内容を行うソフトウェアと考える方が妥当である。また複雑な処理内容を有するソフトウェアは、複数のサブルーチンやワークエリアにて構成されることが一般的であるので、個々のサブルーチンやワークエリアが独立した構成要素と考えるべきである。しかしこれらのサブルーチンやワークエリアには、既存のオペレーティングシステムがライブラリ化しているものも多くあり、それらは説明に値しないものが多いことから、その説明を避けるものとし、制御ブロック処理装置の機能を果たすためにどのような機能を果たすサブルーチン、ワークエリアを構成要素として開発すべきかを論点とした説明を行う。

【0049】（第1実施形態）以下、図面を参照しながら

ら第1実施形態について説明する。図1は、オペレーティングシステム（以後OSと記述する）を搭載したシステムの一例を示したブロック図である。ブロック図における矢印は処理の遷移を示し、破線の矢印はデータの遷移を示す（以下、ブロック図における矢印は同様の意味とする）。このシステムはCPUによるプログラム制御装置101、入力装置102、出力装置103、メモリ装置104より構成される。更にプログラム制御装置101は、入力装置102及び出力装置103との入出力を制御する入出力制御111、入出力制御より入力を受け付ける割り込み処理部112、プログラムの機能を幾つかに分割したタスク113、タスク113の実行順序を制御するOS114から構成される。

【0050】プログラム制御装置101は、メモリ装置104に記憶されている処理プログラムにしたがってプログラムの実行を行う。OS114は、制御ブロック処理装置に相当するものであり、タスク113に付加した優先度やタスクステータスに基づき実行タスクを切り換えることにより、タスク113の実行順序を制御する。優先度やタスクステータスは、割り込み処理部112やタスク113がOS114の提供するシステムコールサービス141を呼び出すことにより変更される。

【0051】入出力制御111は入力装置102より受け付けた入力事象を外部割り込みとして割り込み処理部112に通知する。通知を受けた割り込み処理部112は、他の制御部が実行中であっても直ちに実行される。システムコールサービス141には、同期制御、排他制御、タスク間通信機能、メモリ管理機能など複数の機能がある。

【0052】図2は、メモリ装置104の外部参照領域に配置されているリンクヘッダ90、リンク構造データの一例を示す図である。本図におけるTCB91は、実行待ち状態にあるタスクのそれぞれに対応するものであり、自身に対応するタスクを実行した後、次に実行すべきタスクのTCBを示すリンク情報94と、このタスクの実行待ち状態が解除されるための条件を示す待ち条件解除情報95とを有する。

【0053】リンクヘッダ90はイベントフラグ管理ブロックとも呼ばれ、複数のTCBのうち、最初に実行すべきTCBのアドレスが設定されるリンク先頭データアドレス記憶領域92と、リアルタイムシステムにおいて成立している事象を示す値が設定される事象フラグ値93とを有する。これらの具備している点において、リンクヘッダ90は従来技術のものと変わらないが、リンクヘッダ90に次回処理対象データアドレス記憶領域105が追加されている点が従来技術と異なる。即ち、次回処理対象データアドレス記憶領域105（従来技術において局所的領域に配置されていた次回処理対象記憶領域に相当するものである。）が本実施形態では、外部参照可能領域内のリンクヘッダ90に配置されているのである。

【0054】図3は、割り込み処理により割り込まれる側の処理におけるシステムコール処理部201の構成を機能的に示すブロック図である。図3に示すように、第1実施形態におけるシステムコールサービス141は、メイン制御部301と、割り込み制御部302と、今回処理対象データアドレス取得部303と、データ処理終了判定部304と、次回処理対象データ決定部305と、データ処理部306と、今回処理対象データアドレス記憶領域310とからなる。

【0055】メイン制御部301は、システムコール処理の開始から終了までの一連の処理を、各制御部を用いて実行する。割り込み制御部302は、外部割り込みの受け付けを禁止したり許可したりする。今回処理対象データアドレス取得部303は、外部参照可能な領域に設置されたリンクヘッダより、リンク先頭データアドレス記憶領域または次回処理対象データアドレス記憶領域105に格納されているアドレス情報を抜き出し、今回処理対象データアドレス記憶領域310に格納する。

【0056】データ処理終了判定部304は、今回処理対象データアドレス記憶領域310の情報より、リンク構造データの処理を継続するか終了するかを決定する。次回処理対象データ決定部305は、今回処理対象から次回処理対象を導き出し、リンクヘッダ90内に存在する次回処理対象データアドレス記憶領域105に求めたデータのアドレスを設定する。

【0057】データ処理部306は、今回処理対象に相当するタスクに対して何らかの処理を実行する。割り込み時におけるシステムコールサービス141の処理を、図3のブロック図と、図4に示すフローチャートを用いて更に詳細に説明する。図4は、割り込み時におけるシステムコールサービス141の処理内容を示すフローチャートである。

【0058】先ずシステムコール処理部にてリンク構造データの処理を開始すると、ステップS402において割り込み制御部302にて割り込みの受け付けを禁止する。次にステップS403では、今回処理対象データアドレス取得部303がリンクヘッダのリンク先頭データアドレス記憶領域からリンクの先頭データアドレスを抜き出し、今回処理対象データアドレス記憶領域310に格納する。今回処理対象データアドレスは、時にリンクの終端を示す情報となる。

【0059】続いてステップS404では、今回処理対象データアドレスを設定後、データ処理終了判定部304が設定された今回処理対象データアドレスがリンクの終端情報か否かを調べる。終端であればリンク構造データの処理を終了し(ステップS404-Yes)、終端でなければ処理を継続する(ステップS404-No)。処理が継続された場合、ステップS405において次回処理対象データ決定部305が今回処理対象データの内部情報から次回処理対象データのアドレスを求め、次回処

理対象データアドレス記憶領域105に格納する。本実施形態のリンク構造データは、データ内のリンク情報でリンクを構成しているため、このリンク情報の内容を次回処理対象データアドレス記憶領域105に格納する。

【0060】ステップS406では、次に、データ処理部306が今回処理対象データに相当するタスクに関するデータ処理を行う。データ処理が完了した後、ステップS407では、割り込み制御部302にて一時的に割り込みの受け付け状態を禁止状態から許可状態に変更し、その後、ステップS408において割り込み処理の受け付けを禁止状態に戻す。

【0061】続いて、ステップS409では、今回処理対象データアドレス取得部303が次回処理対象データアドレス記憶領域105に格納されていた次回処理対象データアドレスを抜き出し、今回処理対象データアドレス記憶領域310に格納する。以後、データ処理終了判定部304にて処理の終了を検知するまでステップS404～ステップS409の処理を繰り返す。なお、リンク構造データの処理はリンクデータの終端を発見する以外に、ステップS406において任意の条件が満たされることにより終了する場合もある。

【0062】図5は、割り込み処理を行うシステムコール処理部201の機能を示すブロック図である。図5に示すように、割り込んだ側の処理を行うシステムコール処理部201は、メイン制御部501と、データ処理部502と、リンク構造データ変更判定部503と、次回処理対象データアドレス初期化部504とからなる。

【0063】メイン制御部501は、割り込み処理から呼び出されたシステムコール処理の開始から終了までの一連の処理を、各制御部を用いて実行する。データ処理部502は、リンク構造データ内のICBに相当する何れかのタスクに対する何らかの処理を実行する。リンク構造データ変更判定部503は、データ処理部502において、リンク構造が変更されたか否かの判定を行う。

【0064】次回処理対象データアドレス初期化部504は、リンク構造データ変更判定部503にてリンク構造が変更されたと判断された場合に、次回処理対象データアドレス記憶領域105にリンク先頭データアドレス記憶領域92の内容を設定する。図5のように構成されたシステムコール処理部201の処理を、図6に示すフローチャートを用いて詳細に説明する。なお、図6のフローチャートは、図4に示す割り込まれた側の処理のフローチャートに、システムコール処理中に割り込んだ他のシステムコール処理のフローチャートを追加したものである。

【0065】図4のフローチャートにおいてステップS407とステップS408との間において、任意のシステムコールにてデータ処理が完了し、一時的に割り込みの受け付けが許可されている。この許可区間において外部割り込みが発生し、呼び出された割り込み処理の中か

らシステムコールが呼び出され、システムコールの処理が開始される。

【0066】割り込んだ側の処理が開始されると、ステップS602においてデータ処理部502が今回処理対象に相当するタスクに関する処理を行う。データ処理では、リンクの任意な位置にTCBを追加したり、リンク内のTCBを解除したり、あるいはリンク構造データとは無関係の処理を実行したりする。タスクに関する処理を行った後、ステップS603においてリンク構造データ変更判定部503によりリンク構造データのリンク情報を変更したか否かが判定され、リンク情報を変更されていなければシステムコール処理を終了（ステップS603→変更無）、リンク情報が変更されていれば処理を継続する（ステップS603→変更有）。

【0067】ステップS604においてリンク情報が変更されていた場合、次回処理対象データアドレス初期化部504により、リンクヘッダ90のリンク先頭データアドレス記憶領域から変更後のリンク構造データのリンク先頭データアドレスを抜き出し、次回処理対象データアドレス記憶領域105に格納する。続いて、リンク構造データの具体例を参照しながら、システムコール処理部201の割り込んだ側の処理、割り込まれた側の処理について図9を参照しながら説明する。図9は、次回処理対象データアドレス記憶領域105が更新されてゆく様子状態(c-1)、状態(c-2)、状態(c-3)、状態(c-4)……・状態(c-7)を示す図である。この具体例は、事象フラグを“1”に設定する旨のシステムコール“set_flg(1)”が呼び出された場合を想定したものである。

【0068】（状態(c-1)）システムコール set_flg(1)が呼び出されたので、システムコールサービス141はリンクヘッダ90における事象フラグ値93に値“1”を設定する。一方、図9に示すように、リンク構造データにはTCB1-TCB2-TCB3-TCB4が配置されており、リンクヘッダ90には、TCB1が設定されている。また事象フラグ値93には“1”が設定されている。更に次回処理対象データアドレス記憶領域105にはTCB1が設定されている。TCB1、TCB2、TCB3、TCB4は、それぞれ事象フラグが“2”、“2”、“1”、“3”に設定されるのを待っている。この状態で、set_flg(1)が発生して事象フラグが“1”に設定されたものとする。ここでset_flg(1)とは、事象フラグが“1”に設定されたことを通知するシステムコールである。この場合、事象フラグが“1”に設定されるのを待っていたTCB3を事象成立待ち状態から解除する必要がある。ステップS401～ステップS402により、TCB1が先ず今回処理対象となる。このようにTCB1が今回処理対象に設定されると、リンク構造データは状態(c-1)に示すようになる。

【0069】（状態(c-2)）今回処理対象が終了ではないので、ステップS404はNoとなり、ステップS405に移行する。ステップS405では、TCB1のリンク情

報94に設定されているTCB2を破線の矢印に示すように次回処理対象データアドレス記憶領域105にコピーする。これにより、次回処理対象はTCB2となる。次回処理対象を設定した後、ステップS406においてタスク1の処理を実行する。ここでもし事象フラグの値と、これから実行しようとしているタスクの待ち条件解除情報95とが一致すれば、当該タスクの事象成立待ち状態は解除される。しかしタスク1の待ち条件解除情報95の値は“2”であり、事象フラグの値は“1”なので不一致となり、TCB1はリンク構造データから解除されることなくリンク構造データに残存する。タスク1の実行をスキップした後、ステップS407に移行する。ステップS407では割り込み処理を許可する。このように割り込み処理を許可した時点で直後に割り込み処理が発生したものと

とする。

【0070】（状態(c-3)）この割り込み処理の発生時には、ステップS601の処理に移行する。ここでステップS602の割り込み処理実行時においてシステムコールrel_wait(task2)が呼出されたものとする。ここでrel_wait(task2)とは、TCB2の事象成立待ち状態を強制解除するシステムコールである。このrel_wait(task2)の実行により、TCB2がリンク構造データから解除されてステップS603に移行する。TCB2がリンク構造データから解除されたので、リンク構造データに変更が生じたことになり、ステップS603がYesとなる。ステップS603がYesとなれば、ステップS604においてリンク先頭データアドレス記憶領域92に示されているTCB1を次回処理対象データアドレス記憶領域105に設定する。この時点で、次回処理対象データアドレス記憶領域105はTCB1と設定される。

【0071】（状態(c-4)）以上の処理で割り込み処理が終了する。割り込み処理の終了により、システムコールset_flg(1)の処理が再開される。割り込み処理が発生したのは、ステップS407であるので、set_flg(1)の処理はステップS408から再開する。ここで割り込み処理におけるステップS408～ステップS409により、TCB1が次回処理対象に設定されたので、リンク構造データの処理をTCB1からやり直す。即ち、ステップS404がYesになるまで、ステップS404～ステップS409が繰り返して行われる。

【0072】ステップS404～ステップS409の繰り返しにより、リンク構造データは最終的に状態(c-4)に示すものとなる。続いて割り込まれた側の処理においてリンク構造データに新たなTCBが挿入された場合の処理の具体例について図10を参照しながら説明する。

（状態(d-1)）図10に示すように、リンク構造データにはTCB2-TCB3-TCB4が配置されており、リンクヘッダ90には、TCB1が設定されている。ここでシステムコールset_flg(1)が呼び出されたので、システムコールサービス141はリンクヘッダ90における事象フラグ

値 93 に値 "1" を設定する。また次回処理対象データアドレス記憶領域 105 には TCB2 が設定されている。TCB2、TCB3、TCB4 間のリンク情報 94 は、各タスクの優先度順に設定されているものとする。ここで優先度は、各タスクの番号に等しいものとし、このタスク番号が小さいもの程、優先度は高いと考える。

【0073】またこれらのタスクは、それぞれ事象フラグが "2", "1", "3" に設定されるのを待っている。この状態で、set_flg(1) が発生して事象フラグが "1" に設定されたものとする。この場合、事象フラグが "1" に設定されるのを待っていた TCB3 を事象成立待ち状態から解除する必要がある。ステップ S401～ステップ S402 により、TCB2 が先ず今回処理対象となる。TCB2 が今回処理対象に設定されると、リンク構造データは状態 (d-1) に示すようになる。

【0074】(状態 (d-2)) 今回処理対象が終了ではないので、ステップ S404 は No となり、ステップ S405 に移行する。ステップ S405 では、TCB2 のリンク情報 94 に設定されている TCB3 を次回処理対象データアドレス記憶領域 105 にコピーする。これにより、次回処理対象は TCB3 となる。次回処理対象を設定した後、ステップ S406 においてタスクの処理を実行する。ここでもし事象フラグの値と、これから実行しようとしているタスクの待ち条件解除情報 95 とが一致すれば、当該タスクの事象成立待ち状態は解除される。しかしタスク 2 の待ち条件解除情報 95 の値は "2" であり、事象フラグの値は "1" なので、不一致となり、TCB2 はリンク構造データから解除されることなくリンク構造データに残存する。タスク 2 の実行をスキップした後、ステップ S407 に移行する。ステップ S407 では割り込み処理を許可する。このように割り込み処理を許可した時点の直後に割り込み処理が発生したものである。

【0075】(状態 (d-3)) この割り込み処理の発生時には、ステップ S601 の処理に移行する。ここでステップ S602 の割り込み処理実行時においてシステムコール wait_flg(task1) が呼出されたものとする。wait_flg(task1) とは、タスク 1 を事象成立待ち状態として新たにリンク構造データに追加するシステムコールであり、この wait_flg(task1) の実行により、TCB1 がリンク構造データに追加されてステップ S603 に移行する。ここで TCB1 がリンク構造データに追加されたので、リンク構造データに変更が生じたことになり、ステップ S603 が Yes となる。ステップ S603 が Yes となれば、ステップ S604 においてリンク先頭データアドレス記憶領域 92 に示されている TCB1 を次回処理対象データアドレス記憶領域 105 に設定する。この時点で、次回処理対象データアドレス記憶領域 105 は TCB1 と設定される。

【0076】(状態 (d-4)) 以上の処理で割り込み処理が終了する。割り込み処理の終了により、set_flg(1) の処理が再開される。割り込み処理が発生したの、ステ

ップ S407 であるので、set_flg(1) の処理をステップ S408 から再開する。ここで割り込み処理におけるステップ S408～ステップ S409 により、TCB1 が今回処理対象に設定されたので、リンク構造データの処理を TCB1 からやり直す。即ち、ステップ S404 が Yes になるまで、ステップ S404～ステップ S409 が繰り返される。

【0077】ステップ S404～ステップ S409 の繰り返しにより、リンク構造データは最終的に状態 (d-4) に示すものとなる。以上のように本実施形態によれば、割り込み禁止時間の短縮化を行う際のシステムコール処理時間の増加を最少限に留めることができるとともに、タスクに付加した優先度の趣旨通りにタスク制御が行われない時間を短縮化することができ、OS の実行時間をあまり考慮する必要はない。これによりリアルタイムシステム構築におけるアプリケーション設計が効率化される。

【0078】(第 2 実施形態) 第 1 実施形態では、次回処理対象記憶領域をリンク先頭データアドレス記憶領域 92 に示されているものに設定していたが、第 2 実施形態では、割り込まれた側の処理において最後に実行されていた TCB の次の TCB から、割り込んだ側の処理を再開するように構成された実施形態である。

【0079】図 7 は、割り込んだ側の処理におけるシステムコール処理部 201 の構成を機能的に示すブロック図である。上記のような再開処理のため、図 7 に示すブロックでは、図 3 に記載したブロック図に、システムコール処理中に割り込んだ他のシステムコール処理のブロック図が追加されている。ただし、図 3 に既に示した部分は省略してある。

【0080】図 7 に示すように、第 2 実施形態におけるシステムコールサービス 141 は、メイン制御部 701 と、処理再開データ位置記憶部 702 と、データ処理部 703 と、リンク構造データ変更判定部 704 と、処理再開データ位置決定部 705 と、処理再開データ位置記憶部 710 とからなる。メイン制御部 701 は、割り込み処理から呼び出されたシステムコール処理の開始から終了までの一連の処理を、各制御部を用いて実行する。

【0081】処理再開データ位置記憶部 702 は、リンクヘッダの次回処理対象データアドレス記憶領域 105 に格納されている次回処理対象データアドレスを処理再開データ位置として、処理再開データ位置記憶領域 710 に格納する。それと共に、この再開対象の優先度と、再開対象のリンク先とを処理再開データ位置記憶領域 710 に保存する。

【0082】データ処理部 703 は、リンク構造データ内の任意の TCB1 に対応するタスクに関する処理を行う。リンク構造データ変更判定部 704 は、データ処理部 703 において、リンク構造が変更されたか否かの判定を

行う。処理再開データ位置決定部705は、リンク構造データ変更判定部704にてリンク構造が変更されたと判断された場合に、次回処理対象データアドレス記憶領域105に処理再開のために最適なリンクデータアドレスを設定する。

【0083】図7のブロック図に示したように構成された第2実施形態に係るシステムコール処理部201の処理を図8に示すフローチャートを用いて更に詳細に説明する。なお、図8のフローチャートは、図4に示した第1実施形態の割り込んだ側の処理のフローチャートに、システムコール処理中に割り込んだ他のシステムコール処理のフローチャートを追加したものである。

【0084】任意のシステムコールにてデータ処理が完了し、一時的に割り込みの受け付けを許可した時点(ステップS407とステップS408の間)で外部割り込みが発生し、呼び出された割り込み処理の中からシステムコールが呼び出され、システムコールの処理を開始する。まず、ステップS802において処理再開データ位置記憶部702がリンクヘッダの次回処理対象データアドレス記憶領域105に格納されている次回処理対象データアドレスを再開対象として、処理再開データ位置記憶領域710に格納する。それと共に、この再開対象の優先度と、再開対象のリンク先とを処理再開データ位置記憶領域710に保存する。

【0085】次に、ステップS803では、データ処理部703が今回処理対象データに関する処理を行う。ここでの処理では、リンクの任意な位置にデータを追加したり、リンク構造データ内のTCBをリンクから解除したり、あるいはリンク構造データとは無関係の処理を実行したりする。ステップS804においてデータ処理を行った後、リンク構造データ変更判定部704によりデータ処理がリンク構造データのリンク情報を変更したか否かを判定し、リンク情報が変更されていない場合はシステムコール処理を終了(804-変更無)、リンク情報が変更されていれば処理を継続する(804-変更有)。

【0086】リンク情報が変更されていた場合、ステップS805では、ステップS804のデータ処理において再開対象がリンク構造データから解除されたかを判定する。そうでないならステップS807に移行するが、そうならばステップS806に移行する。ステップS806では、ステップS804のデータ処理において、タスクの事象成立待ち状態を解除する目的のシステムコールが実行され、再開対象が解除された場合、保存されていた再開対象のリンク先を次回処理対象として設定する。

【0087】ステップS807では、ステップS804のデータ処理においてリンク構造データにタスクが挿入されたかを判定する。ステップS808では、再開対象と新たに挿入されたタスクの優先度のどちらが高いかを判定する。再開対象の優先度が高いならステップS81

0に移行するが、新たに挿入されたタスクの優先度が高いならばステップS809に移行する。

【0088】ステップS809では、新たに挿入されたタスクを次回処理対象として設定する。ステップS810では、再開対象を次回処理対象として設定する。続いて図8のフローチャートにおける処理の一例について、図11の説明図を参照しながら説明する。図11は、リンク構造データが図8のフローチャートに示す処理により変更される様子を示す図である。

【0089】(状態(c-1)) 図11に示すように、リンク構造データにはTCB1-TCB2-TCB3-TCB4が配置されており、リンクヘッダ90には、TCB1が設定されている。また事象フラグ値93には"1"が設定されている。また次回処理対象データアドレス記憶領域105にはTCB1が設定されている。TCB1、TCB2、TCB3、TCB4は、それぞれ事象フラグが"2","2","1","3"に設定されるのを待っている。この状態で、set_flg(1)が発生して事象フラグが"1"に設定されたものとする。この場合、事象フラグが"1"に設定されるのを待っていたTCB4を事象成立待ち状態から解除する必要がある。ステップS401-ステップS402により、TCB1が先ず今回処理対象となる。このようにTCB1が今回処理対象に設定されると、リンク構造データは状態(c-1)に示すようになる。

【0090】今回処理対象が終了ではないので、ステップS404はNoとなり、ステップS405に移行する。ステップS405では、TCB1のリンク情報94に設定されているTCB2を次回処理対象データアドレス記憶領域105にコピーする。これにより次回処理対象はTCB2となる。次回処理対象を設定した後、ステップS406においてタスク1の処理を実行しようとする。ここでもし事象フラグの値と、これから実行しようとしているタスクの待ち条件解除情報95とが一致すれば、当該タスクの事象成立待ち状態は解除される。しかしタスク1の待ち条件解除情報95の値は"2"であり、事象フラグの値は"1"なので、不一致となり、TCB1はリンク構造データから解除されることなくリンク構造データに残存する。タスク1の実行をスキップした後、ステップS407に移行する。ステップS407では割り込み処理を許可する。このように割り込み処理を許可した時点の直後に割り込み処理が発生したものととする。

【0091】(状態(c-2)) この割り込み処理の発生時には、ステップS801の処理に移行する。ここでステップS802において再開対象にTCB2を設定し、この再開対象の優先度と、再開対象のリンク先となるTCB3とを保存する。再開対象を設定し、再開対象の優先度及び再開対象のリンク先の保存後、ステップS803において割り込み処理を実行する。この割り込み処理実行時においてシステムコールrel_wait(task2)が呼び出されたものとする。このrel_wait(task2)の実行により、TCB2がリンク構造データから解除されてステップS804に移

行する。TCB2がリンク構造データから解除されたので、リンク構造データに変更が生じたことになり、ステップS804がYesとなる。ステップS804がYesとなれば、ステップS805に移行する。rel_wait(task2)により解除されたのはTCB2なので、rel_wait(task2)により、再開時に次回処理対象となるTCBが消滅してしまったことになる。そのため、ステップS805がYesとなりステップS806に移行する。ここで再開対象は解除されたが、この再開対象のリンク先であったTCB3は、ステップS802において保存されている。ステップS806では、このリンク先であるTCB3を次回処理対象として矢印a1に示すように設定する。

【0092】(状態(c-3))以上の処理で割り込み処理が終了する。割り込み処理の終了により、set_flg(1)の処理が再開される。割り込み処理が発生したのは、ステップS407であるので、set_flg(1)の処理はステップS408から再開する。ここで割り込み処理におけるステップS408～ステップS409により、TCB3が次回処理対象に設定されたので、リンク構造データの処理をTCB3からやり直す。即ち、ステップS404がYesになるまで、ステップS404～ステップS409が繰り返される。

【0093】尚、リンク構造データから解除されたTCBが再開対象以外である場合、次回処理対象の設定は行わない。続いて図12のフローチャートの処理の一例について、図12の説明図を参照しながら説明する。図12は、リンク構造データが図8のフローチャートに示す処理により変更される様子を示す図である。

【0094】(状態(d-1))リンク構造データにはTCB2-TCB3-TCB4が配置されており、リンクヘッダ90のリンク先頭データアドレス記憶領域92には、TCB2が設定されている。また事象フラグ値93には"1"が設定されている。また次回処理対象データアドレス記憶領域105にはTCB3が設定されている。TCB2、TCB3、TCB4は、それぞれ事象フラグが"2"、"1"、"3"に設定されるのを待っている。この状態で、set_flg(1)が発生して事象フラグが(1)に設定されたものとする。この場合、事象フラグが"1"に設定されるのを待っていたTCB3を事象成立待ち状態から解除する必要がある。ステップS401～ステップS402により、TCB1が先ず次回処理対象となる。このようにTCB1が今回処理対象に設定されると、リンク構造データは状態(d-1)に示すようになる。

【0095】今回処理対象が終了ではないので、ステップS404はNoとなり、ステップS405に移行する。ステップS405では、TCB1のリンク情報94に設定されているTCB2を次回処理対象データアドレス記憶領域105にコピーする。これにより、次回処理対象はTCB2となる。次回処理対象を設定した後、ステップS406においてタスク2の処理を実行する。ここでもし事象フラグの値と、これから実行しようとしているタスクの待ち

条件解除情報95とが一致すれば、当該タスクの事象成立待ち状態は解除される。しかしタスク2の待ち条件解除情報95の値は"2"であり、事象フラグの値は"1"なので、不一致となり、TCB1はリンク構造データから解除されることなくリンク構造データに残存する。タスク2の実行をスキップした後、ステップS407に移行する。ステップS407では割り込み処理を許可する。このように割り込み処理を許可した時点の直後に割り込み処理が発生したものとす。

【0096】(状態(d-2))この割り込み処理の発生時には、ステップS801の処理に移行する。ここでステップS802において再開対象にTCB2を設定し、この再開対象の優先度と、リンク先(TCB3)とを保存する。再開対象の設定及び再開対象のリンク先の保存後、ステップS803において割り込み処理を実行する。この割り込み処理実行時においてシステムコールwait_flg(task1)が呼出されたものとする。このwait_flg(task1)の実行により、TCB1がリンク構造データに追加されてステップS804に移行する。TCB1が追加されたので、リンク構造データに変更が生じたことになり、ステップS804がYesとなる。ステップS804がYesとなれば、ステップS805に移行する。ここではTCBが挿入されたのでステップS805はNoとなりステップS807に移行する。ステップS807がYesとなってステップS808に移行する。ステップS808では、再開対象と、新たに挿入されたタスクとの優先度を比較する。ここで、新規に挿入されたTCB1の方が優先度が高いので、新規に挿入されたTCBは再開対象以前に配置されたことがわかる。優先度がより高いTCBが挿入されたので、リンク構造データにおいて新規に挿入されたTCBから処理をやり直すべく、ステップS809では、挿入対象であるTCB1を次回処理対象として矢印a2に示すように設定する。

【0097】(状態(d-3))以上の処理で割り込み処理が終了する。割り込み処理の終了により、割り込まれた側の処理が再開される。割り込み処理が発生したのは、ステップS407であるので、ステップS408から再開する。ここで割り込み処理におけるステップS408～ステップS409により、TCB1が次回処理対象に設定されたので、リンク構造データの処理をTCB1からやり直す。即ち、ステップS404がYesになるまで、ステップS404～ステップS409が繰り返される。

【0098】上記の処理において、再開対象より前にTCBが挿入されたなら、その新たに挿入されたTCBが次回処理対象に設定される。また、再開対象より後にTCBが挿入されたなら、次回処理対象を変更する必要はない。以上のように本実施形態によれば、割り込まれた側の処理において最後に実行されていたTCBの次のTCBから、割り込まれた側の処理を再開するので、割り込まれた側の処理における処理の進捗を向上させることができる。

【0099】(第3実施形態)第2実施形態ではリンク

構造データにおいてTCBが結合されていたが、第3実施形態ではメモリ管理ブロックがリンク構造データに結合されている。ここでリンク構造データに結合されるメモリ管理ブロックは、タスクに割り当てられていないフリーのメモリブロックを示すものである。即ち、フリーメモリブロックは、サイズが小さい順にリンク構造データにおいて配置され、フリーブロックを示すものとして、タスクに割り当てられるのを待つ。

【0100】この状態で、何れかのタスクがget_blkを発行すれば、このget_blkに示される要求を満たすサイズのフリーメモリブロックのうち、最小のものをリンク構造データから探索して、発行元のタスクに割り当てる。図13(a)(b)は第3実施形態におけるメモリ管理ブロックの割り当て処理の一例を示す図である。図13(a)(b)は第3実施形態におけるメモリ管理ブロックの割り当て処理の一例を示す図である。

【0101】図13(a)に示すようにサイズ"350"のフリーメモリブロックを求めるget_blkが何れかのタスクから発行された場合、メモリ管理ブロックBLK1→メモリ管理ブロックBLK2→メモリ管理ブロックBLK3→メモリ管理ブロックBLK4の順で探索が行われ、メモリ管理ブロックBLK4が割り当て対象となる。このようにメモリ管理ブロックBLK4を割り当て対象とすることが確定した段階で割り込み処理が発生したものとす。この割り込み処理において、rel_blk(350)が発行されたものとす。このrel_blk(350)は、サイズ"350"のフリーメモリブロックを開放する旨のシステムコールであり、このシステムコールが実行されたことにより、リンク構造データは図13(b)に示すものとなる。図13(b)を参照すると、メモリ管理ブロックBLK3とメモリ管理ブロックBLK4との間にメモリ管理ブロックBLK5が挿入されていることがわかる。このようにフリーメモリブロックが挿入されれば、第2実施形態と同様、新規に挿入されたフリーメモリブロックと、再開対象として保存したフリーメモリブロックとの優先度を比較し、再開対象を更新すべきか否かを決定する。ここでメモリ管理ブロックBLK4のサイズは"400"であり、メモリ管理ブロックBLK5のサイズは"350"である。これらのサイズを優先度と考えて比較を行えば、メモリ管理ブロックBLK5の優先度が高いことがわかる。そのため、ステップS809において、メモリ管理ブロックBLK5を再開対象に設定して、割り込み処理を終了し、割り込まれた側の処理を再開する。

【0102】第2実施形態の手順で次回処理対象を設定したが、第1実施形態同様、リンクヘッダ90のリンク先頭データアドレス記憶領域92に設定されているフリーメモリブロックを次回処理対象データアドレス記憶領域105に設定してもよい。尚第1〜第3実施形態において説明したオペレーティングシステムをコンピュータ読み取り可能な記録媒体に記録して流通・販売の対象にしても良い。このような記録媒体には、ICカードや光デ

ィスク、フロッピーディスク等があるが、これらに記録された機械語プログラムはプロセッサ、メモリを有する機器にインストールされることにより利用に供される。この機器は、インストールした機械語プログラムを逐次実行して、本実施形態に示した制御ブロック処理装置の機能を実現するのである。

【0103】

【発明の効果】以上説明したように本発明は、リンク情報を用いて結合されている複数の制御ブロックからなるリンク構造データと、複数の制御ブロックのうち何れかの制御ブロックのアドレスが書き込まれるアドレス記憶領域と、アドレス記憶領域にアドレスが書き込まれている制御ブロックを今回処理対象として確定する動作と、その制御ブロックのリンク先の制御ブロックを次回処理対象として確定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込む動作とをリンク構造データにおける全ての制御ブロックについて繰り返す処理対象確定手段と、今回処理対象が確定される度に、この今回処理対象についての処理を行う処理手段と、今回処理対象についての処理が行われてから、次の今回処理対象が確定されるまでの期間において割り込み処理の発生を許可する許可手段とを備え、前記処理対象確定手段は、割り込み処理を許可した期間において割り込み処理が発生した場合に起動され、リンク構造データにおける何れかの制御ブロックを割り込み処理における今回処理対象として確定すると共に、別の制御ブロックを割り込まれた側の処理を再開する際に処理すべき次回処理対象として確定して、当該次回処理対象のアドレスをアドレス記憶領域に書き込むリエントリールーチンを有し、割り込み処理発生後割り込まれた側の処理を再開する場合、及び、割り込み処理が発生しなかった場合において、アドレス記憶領域に書き込まれている制御ブロックを今回処理対象として確定するように構成されている。

【0104】割り込み処理発生後割り込まれた側の処理を再開する場合、割り込み処理が発生しなかった場合の双方において共通の手順で次回処理対象を確定することができる。次回処理対象の確定手順は共通化され、簡略化されるので、特開平9-311794に記載された技術のように、割り込まれた側の処理でブラグの監視を行う必要がなくなる。よって、今回処理対象→次回処理対象の確定に要する時間を最少限に留めることができる。次回処理対象確定に要する時間が短縮されれば、制御ブロックに対応するタスクに割り当て可能なシステム稼動時間が長くなる。タスクに割り当てられたシステム稼動時間が長くなれば、その分だけ、システム設計者の負担は軽減されるので、アプリケーションプログラムを効率良く設計することができる。

【0105】また上記装置において、前記複数の制御ブロックのうち、何れかのものは、割り込み処理実行時において、リンク構造データから解除され、前記リエン

ラントルーチンは、解除されずにリンク構造データに残った制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。

【0106】割り込み処理実行時において何れかの制御ブロックがリンク構造データから解除された場合、割り込んだ側の処理において次回処理対象が正常に設定されるので、割り込まれた側の処理ではフラグを監視する必要はない。割り込まれた側の処理におけるフラグの監視負担が軽減するので、不要なフラグ監視による処理時間の増加によりタスク切り換えまでの時間が軽減させることができ、アプリケーションに相当するタスクが優先度どおり制御されない時間を減少させることができる。

【0107】また上記装置において、割り込み処理は複数の事象のうち何れかが制御ブロック処理装置において成立した際に発生し、制御ブロックのそれぞれには、タスクが対応しており、制御ブロックには、これに対応するタスクが成立を待っている事象を示す事象情報を有しており、前記複数の制御ブロックのうち、その事象情報が割り込み処理において通知された事象と一致したものは、割り込み処理実行時において、実行された後にリンク構造データから解除され、前記リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。このように構成されれば、事象が成立してから、即座に、その事象の成立を待っていたタスクに対して制御を行うことができ、即時応答性を向上させることができる。これにより、リアルタイムシステムを簡易に構成することができる。

【0108】また上記装置において、前記リエントラントルーチンは、リンク構造データにおいて解除された制御ブロックのリンク先となる制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。特開99-311794に記載された技術のように、割り込まれたシステムコール処理においてリンク構造データの先頭から再度処理を行う必要がなくなるため、割り込みによるシステムコール処理時間の増加を最少限に留めることができる。

【0109】また上記装置において、割り込み処理実行時において、リンク構造データには何れかのタスクに対応する制御ブロックが挿入され、前記リエントラントルーチンは、割り込まれた側への再開時に実行すべき再開対象の制御ブロック及び挿入された制御ブロックに対応するタスクの何れかを先に実行すべきかを決定する決定部を備え、決定部により決定された制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。

【0110】割り込まれた側の処理を再開する際に、割り込んだ側の処理側リンク構造データに挿入されたタ

スク制御ブロックを先に処理すべきか、割り込まれた側の処理発生前に次に実行されるべきであったタスク制御ブロックを先に処理すべきかを柔軟に決定されるので、リアルタイムシステムにおける処理効率を向上させることができる。

【0111】また上記装置において、制御ブロックには、それに対応するタスクの優先度に基づいた優先度が付与されており、前記決定部は、割り込まれた側への再開時に実行すべき再開対象の制御ブロックと、挿入された挿入対象の制御ブロックとで優先度を比較して、何れに対応するタスクの何れかを先に実行すべきかを決定するように構成しても良い。割り込まれた側の処理を再開する際に、割り込んだ側の処理側でリンク構造データに挿入されたタスク制御ブロックを先に処理すべきか、割り込まれた側の処理発生前に次に実行されるべきであったタスク制御ブロックを先に処理すべきかを決定するに際して、予め定められた優先度が参照されるので、優先度通りにシステムを動作させることができる。

【0112】また上記装置において、リンク構造データは、複数のタスクのうち、最初に実行すべきタスクに対応する制御ブロックを示すリンクヘッダを有しており、リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックのうち、リンクヘッダに示される制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。割り込み処理から呼び出されたシステムコール処理においてリンク構造データの整合性を保つために必要な前記外部参照可能なメモリに設定するデータの選定処理が不要となるため、割り込み処理の高速化が図れる。

【0113】また上記装置において、制御ブロック処理装置は、複数のタスクと、空きのメモリブロックであるフリーメモリブロックとを有し、各制御ブロックは、複数のフリーメモリブロックのそれぞれと対応しており、前記処理手段は、今回処理対象となる制御ブロックに割り当てるフリーメモリブロックを、何れかのタスクに割り当てる割当部を備えるように構成しても良い。このように構成した場合、フリーメモリブロックを何れかのタスクに効率良く割り当てることができる。

【0114】また上記装置において、前記複数の制御ブロックのうち、何れかのものは、割り込み処理実行時において、リンク構造データから解除され、前記リエントラントルーチンは、解除されずにリンク構造データに残った制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。割り込み処理実行時において何れかのメモリ制御ブロックがリンク構造データから解除された場合、割り込んだ側の処理において次回処理対象が正常に設定されるので、割り込まれた側の処理はフラグ情報を監視する必要はない。割り込まれた側の処理におけるフラグ情

報の監視負担が軽減するので、不要なフラグ監視による処理時間の増加によりタスク切り換えまでの時間が軽減するので、アプリケーションに相当するタスクが優先度どおり制御されない時間を減少させることができる。

【0115】また上記装置において、前記リエントラントルーチンは、リンク構造データにおいて解除された制御ブロックのリンク先となる制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。特開平9-311794に記載された技術のように、割り込まれたシステムコール処理においてリンク構造データの先頭から再度処理を行う必要がなくなるため、割り込みによるシステムコール処理時間の増加を最少限に留めることができる。

【0116】また上記装置において、前記割り込み処理実行時において、リンク構造データには何れかのフリーメモリブロックに対応する制御ブロックが挿入され、前記リエントラントルーチンは、割り込まれた側の処理の再開時に割り当てるべき再開対象の制御ブロック及び挿入された制御ブロックに対応するフリーメモリブロックの何れを先にタスクに割り当てるべきかを決定する決定部を備え、決定部により決定された制御ブロックを次回処理対象として確定して、そのアドレスをアドレス記憶領域に書き込むように構成しても良い。割り込まれた側の処理を再開する際に、割り込んだ側の処理側でリンク構造データに挿入されたメモリ制御ブロックを先にタスクに割り当てるべきか、割り込まれた側の処理発生前に次に実行されるべきであったメモリ制御ブロックを先にタスクに割り当てるべきかが柔軟に決定されるので、リアルタイムシステムにおける処理効率を向上させることができる。

【0117】また上記装置において、制御ブロックには、それに対応するフリーメモリブロックのメモリサイズに基づいた優先度が付与されており、前記決定部は、割り込まれた側への再開時にタスクに割り当てるべき再開対象の制御ブロックと、挿入された挿入対象の制御ブロックとで優先度を比較して、何れに対応するフリーメモリブロックの何れを先にタスクに割り当てるべきかを決定するように構成しても良い。割り込まれた側の処理を再開する際に、割り込んだ側の処理側でリンク構造データに挿入されたメモリ制御ブロックを先にタスクに割り当てるか、割り込まれた側の処理発生前に次にタスクに割り当てるべきであったメモリ制御ブロックを先に処理すべきかを決定するに際して、フリーメモリブロックのサイズが参照されるので、サイズの順序でフリーメモリブロックを各タスクに割り当てることができる。

【0118】また上記装置において、リンク構造データは、複数のフリーメモリブロックのうち、最初にタスクに割り当てるべきフリーメモリブロックに対応する制御ブロックのアドレスを示すリンクヘッダを有しており、第1割当部は、解除されずリンク構造データに残った

制御ブロックのうち、リンクヘッダに示される制御ブロックのアドレスを大域的領域に書き込むように構成しても良い。割り込み処理から呼び出されたシステムコール処理においてリンク構造データの整合性を保つために必要な前記外部参照可能なメモリに設定するデータの選定処理が不要となるため、割り込み処理の高速化が図れる。

【図面の簡単な説明】

【図1】オペレーティングシステムを搭載したコンピュータ装置の一例を示したブロック図である。

【図2】メモリ装置104の外部参照領域に配置されているリンクヘッダ90、リンク構造データの一部を示す図である。

【図3】割り込み処理により割り込まれる側の処理におけるシステムコール処理部201の構成を機能的に示すブロック図である。

【図4】割り込み処理により割り込まれる側の処理におけるシステムコール処理部201の処理内容を示すフローチャートである。

【図5】割り込み処理を行うシステムコール処理部201の機能をj示すブロック図である。

【図6】図4に示す割り込まれた側の処理のフローチャートに、システムコール処理中に割り込んだ他のシステムコール処理のフローチャートを追加した図である。

【図7】割り込んだ側の処理におけるシステムコール処理部201の構成を機能的に示すブロック図である。

【図8】図4に示した第1実施形態の割り込んだ側の処理のフローチャートに、システムコール処理中に割り込んだ他のシステムコール処理のフローチャートを追加した図である。

【図9】次回処理対象データアドレス記憶領域105が更新されてゆく様子状態(c-1)、状態(c-2)、状態(c-3)、状態(c-4)を示す図である。

【図10】リンク構造データにはTCB2-TCB3-TCB4が配置されており、リンクヘッダ90には、TCB1が設定されている状態を示す図である。

【図11】リンク構造データが図8のフローチャートに示す処理により変更される様子を示す図である。

【図12】リンク構造データが図8のフローチャートに示す処理により変更される様子を示す図である。

【図13】第3実施形態におけるフリーメモリブロックの割り当て処理の一例を示す図である。

【図14】OSの基本機能を説明するためのブロック図であり、タスクの実行順序管理、フリーメモリブロックの割当順序管理に直接関係する機能ブロックを示す図である。

【図15】リエントラントルーチンを説明するための図である。

【図16】リンクヘッダ90と、複数のTask Control Block (TCB)からなるリンク構造データの一部を示す図で

ある。

【図17】図11に示したリンク構造データに基づいて、今回処理対象記憶領域、次回処理対象記憶領域が更新されてゆく様子(状態(a-1)、状態(a-2)、状態(a-3)、状態(a-4)……状態(a-7))を示す図である。

【図18】リンク構造データに示される複数のタスクが順次実行されてゆく様子を示す図である。

【図19】割り込み処理発生時の許可区間、禁止区間を設けた図である。

【図20】割り込み処理発生時のメモリ領域の割り当てを示す図である。

【図21】割り込み処理許可区間において割り込み処理が発生し、この割り込み処理において事象フラグを“2”に設定する旨のシステムコール“set_flg(2)”が呼び出された場合に今回処理対象記憶領域、次回処理対象記憶領域がどのように変化するかを示す図である。

【図22】割り込み処理許可区間において割り込み処理が発生し、この割り込み処理においてシステムコール“set_flg(2)”が呼び出された場合にリンク構造データがどのように変化するかを示す図である。

【図23】割り込み処理許可区間において割り込み処理が発生し、この割り込み処理においてシステムコールサービス set_flg(2)が呼び出された場合に今回処理対象記憶領域、次回処理対象記憶領域がどのように変化するかを示す図である。

【図24】割り込み処理許可区間において割り込み処理が発生し、この割り込み処理においてシステムコールサービス set_flg(2)が呼び出された場合にリンク構造データと、フラグ情報とがどのように変化するかを示す図である。

【符号の説明】

90 リンクヘッダ

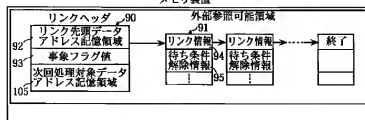
92 リンク先頭データアドレス記憶領域

* 93	事象フラグ値
94	リンク情報
95	条件解除情報
101	プログラム制御装置
102	入力装置
103	出力装置
104	メモリ装置
105	次回処理対象データアドレス記憶領域
111	入出力制御
112	処理部
113	タスク
141	システムコールサービス
201	システムコール処理部
202	タスクスケジューラ
203	ディスパッチャ
301	メイン制御部
302	割り込み制御部
303	今回処理対象データアドレス取得部
304	データ処理終了判定部
20 305	次回処理対象データ決定部
306	データ処理部
310	今回処理対象データアドレス記憶領域
501	メイン制御部
502	データ処理部
503	リンク構造データ変更判定部
504	次回処理対象データアドレス初期化部
701	メイン制御部
702	処理再開データ位置記憶部
703	データ処理部
30 704	リンク構造データ変更判定部
705	処理再開データ位置決定部
710	処理再開データ位置記憶領域

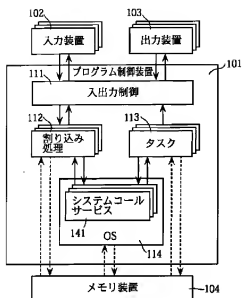
*

【図2】

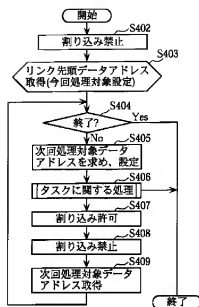
メモリ装置



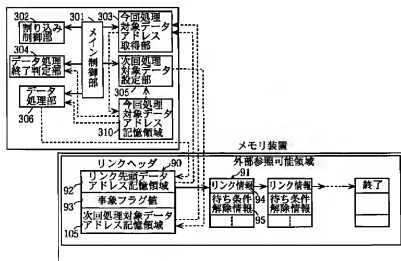
【図1】



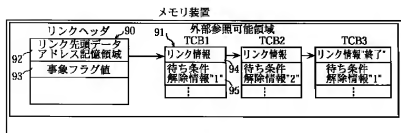
【図4】



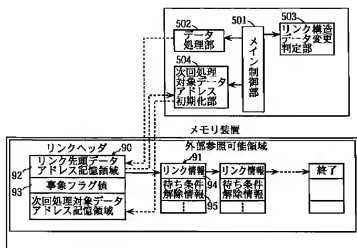
【図3】



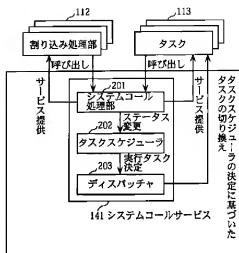
【図16】



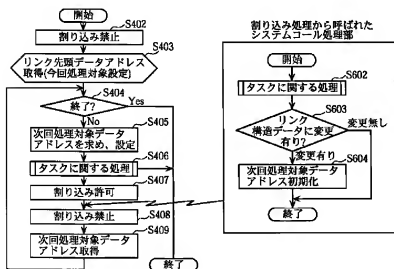
【図5】



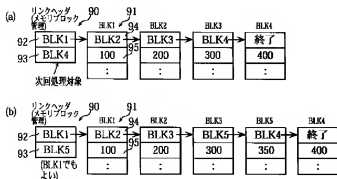
【例 14】



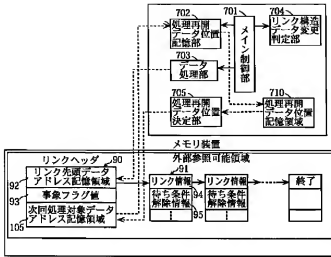
【図6】



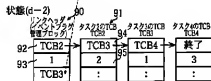
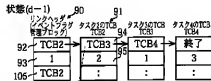
【图 13】



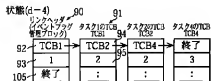
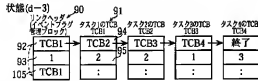
【図7】



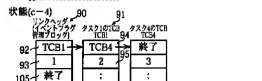
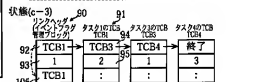
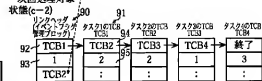
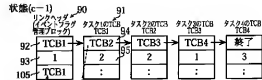
【図10】



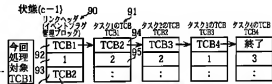
システムコールwait_sig(task)が呼び出された



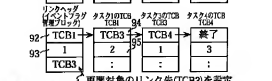
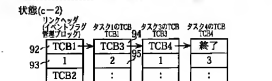
【図9】



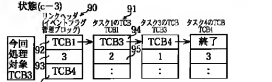
【図11】



105 割り込みが発生した場合、再開対象になる
割込処理発生 再開対象TCB2と、再開後のリンク先TCB3と、再開後の優先度とを格納する
割込処理における rel_wait(task2)が実行される(TCB2が
システムコール リンク構造テーブルを参照)

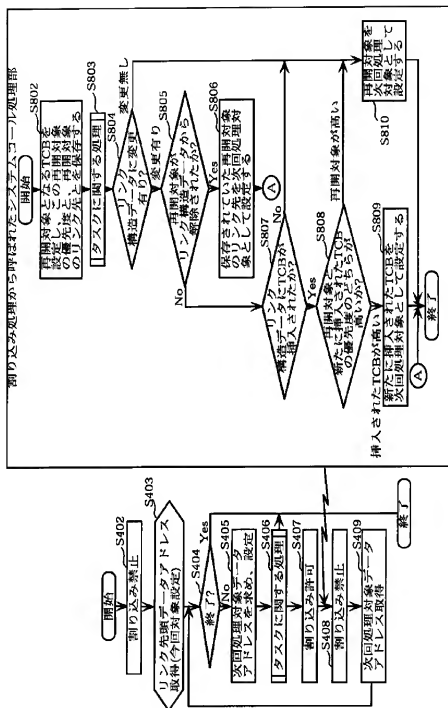


再開対象のリンク先(TCB3)を決定

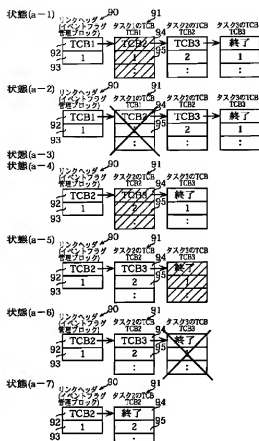


105

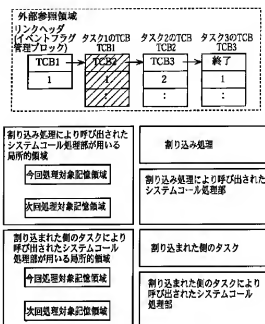
【図8】



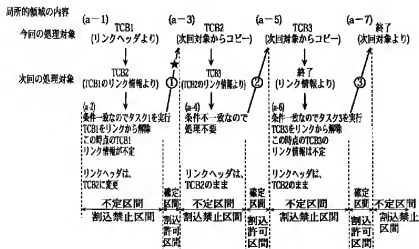
【図18】



【図20】



【図19】



【図23】

